MAGAZINE VOOR DE ACTIEVE MSX GEBRUIKER

MSX®

COMPUTER MAGAZINE

MSX 38

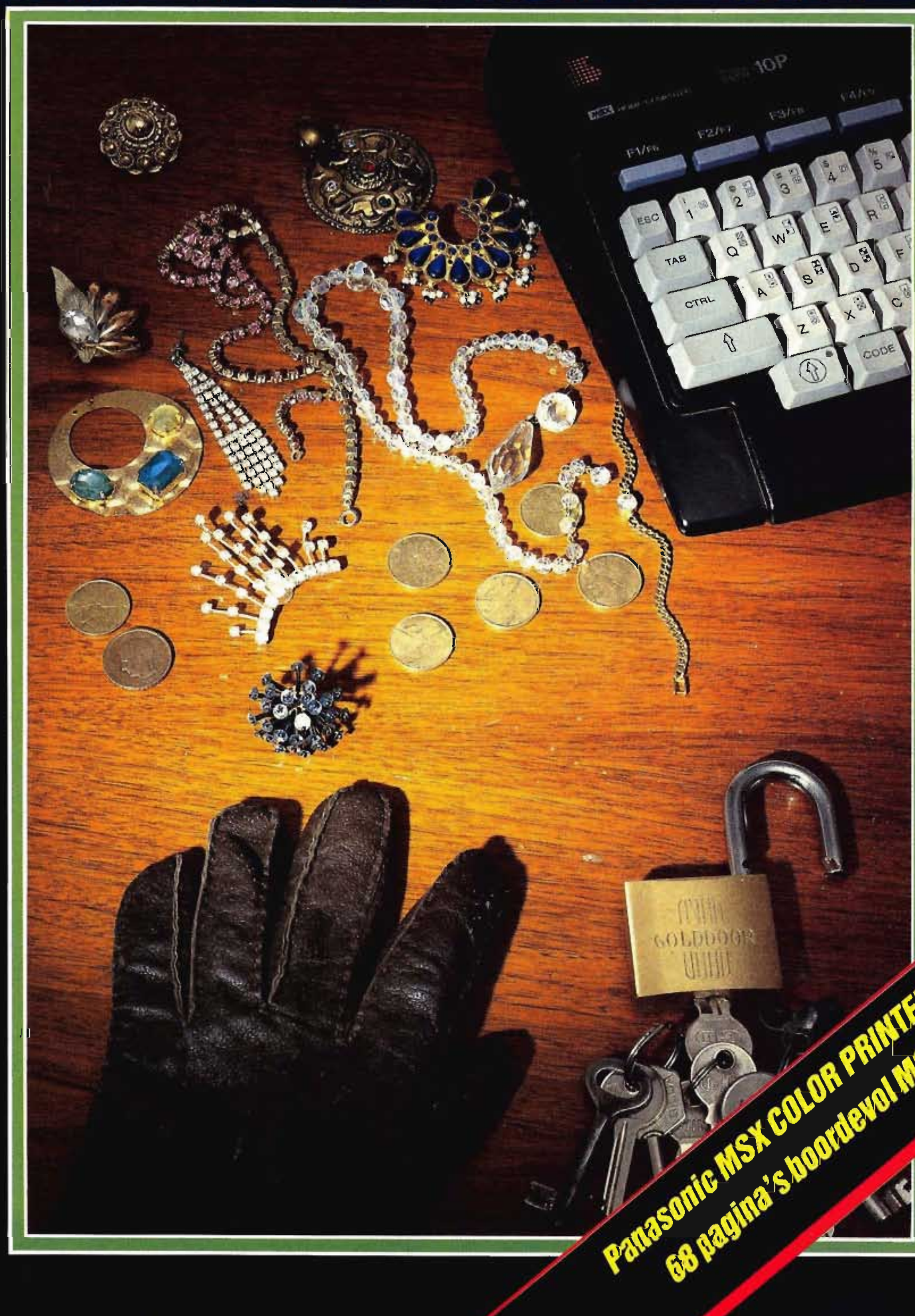
Bug in AackoScribe verholpen
Tokens enzo: Basic intern
Het CMD-commando, onbekend gebied
De trukendoos
Lezers Helpen Lezers
Kort & Krachtig

Listings:
SCRSAV, spaart uw monitor
List-Extra
ZAS-BAS:
Z80-assembler in Basic

Spellen:
De nieuwste Japanse importen
EHBO: spellentips

Tests:
EMCIEBIECIE, MSX Compiler?
De MSX-insteekkaart voor PC
NMS 1436 printer

5e JAARGANG – NR. 37
MEI 1990
ƒ 6,95/BFR 140



Panasonic MSX COLOR PRINTER!
68 pagina's boordevol MSX!

Nabestellen kan ook!

In ieder nummer van MSX Computer Magazine vindt u artikelen over tal van onderwerpen. Vaak zal blijken dat we al geschreven hebben over iets dat uw interesse heeft. In dat geval loont het de moeite het betreffende nummer na te bestellen. Daarom publiceren we regelmatig een overzicht van de artikelen die reeds in MCM verschenen zijn. Hierbij geven we aan of een artikel voor MSX (») of MS-DOS(●) bestemd is. Staat er niets voor dan is het artikel algemeen van aard. Losse nummers kosten ƒ 6,95 per stuk. Dit is exclusief porto- en administratiekosten. Die worden namelijk evenredig met het aantal door u bestelde exemplaren berekend. De nummers 1, 2, 4, 5, 6, 9 ,22 en 23 zijn niet meer voorradig. Wilt u artikelen uit deze nummers nalezen dan kunt u daarvan kopieen bestellen. De kosten hiervoor zijn ƒ 0,25 per pagina, plus ƒ 2,50 voor porto- en administratiekosten. Nabestellingen kunt u doen door de door u verlangde nummers of artikelen te vermelden op een briefkaart en die te sturen naar: Wegener Tijl Tijdschriften Groep B.V. , Afdeling Lezersservice, Postbus 9943, 1006 AP, Amsterdam. De door u bestelde nummers/artikelen ontvangt u thuis samen met een acceptgirokaart voor de betaling.

COMPUTERS

DISKDRIVES

PRINTERS

COMMUNICATIE

MONITOREN

DIV. HARDWARE

LOSSE NUMMER SERVICE

LISTINGS

EDUCATIEF

TOEPASSINGEN

UTILITY'S/TALEN

MSX COMPUTER MAGAZINE
is een uitgave van
Wegener Tijl Tijdschriften Groep b.v., Amsterdam

Hoofdredacteur
Wammes Witkop

Medewerkers
Max Barber, Paul te Bokkel, Mirjam Engelberts, Hans Niepoth, Harry van Horen, Markus The, Mariëlle Mink, André Knip, Edgar Hildering, Robbert Wethmar, Lies Muller, Mathijs Perdec, Kees Reedijk, Aat van Uijen, Wim Vredevoogd.

Redactie–adres
Postbus 75142, 1070 AC, Amsterdam
Fax: 020–862719

Vragentelefoon redactie
De redactie is telefonisch alleen bereikbaar via 020–860743. Op dit nummer staat een antwoordapparaat, waarop we eventuele correcties op artikelen en listings inspreken. Bovendien zijn we minimaal één keer per week via dit nummer rechtstreeks bereikbaar. Wanneer, dat maken we ook via het antwoordapparaat bekend.

Abonnementen
Wegener Tijl Tijdschriften Groep B.V.
Postbus 9943, 1006 AP Amsterdam
Telefoon 06-0224222, telefax 020-5182843

Abonnementsprijs
Een abonnement op MCM kost ƒ 50,– (8 nummers)
Zie de bon elders in het blad.
Het abonnement kan elk gewenst moment ingaan en wordt automatisch verlengd, tenzij twee maanden voor de vervaldatum schriftelijk opgezegd.

Advertenties
Herman de Haan
tel.: 020–845975
Geldend advertentie tarief 1 januari 1990

Programmaservice
De programmaservice is verhuisd. Voor het bestellen van cassettes en diskettes kunt u de bon elders in dit blad invullen en opsturen naar:
MCM LezersService
Postbus 75142
1070 AC Amsterdam
Voor vragen: 020-860743
Zie pagina 4 voor uitgebreide informatie.

Losse nummers
Heeft u vragen over het nabestellen van losse nummers, bel dan met Adrie Donkervoort, afdeling lezersservice 020-5182828

Vormgeving
Mariëlle Mink

Cover–foto
Jan Bartelsman

Distributie
Beta Press/van Ditmar, Burg. Krollaan 14,
5126 PT, Gilze

Verschijning
MSX Computer Magazine verschijnt acht maal per jaar

Toezenden materiaal
Tenzij uitdrukkelijk anders overeengekomen heeft MCM het recht om vrijelijk te beschikken over alle haar toegezonden materiaal.
Terugzending van ongevraagd toegezonden materiaal zal alleen plaatsvinden als er een geadresseerde en voldoende gefrankeerde retour–enveloppe is bijgesloten

Inhoud MSX Computer Magazine 38

Rubrieken

Besprekingen

Listings

Neem een abonnement

Wordt nu abonnee van MSX Computer Magazine en u ontvangt geheel gratis het 160 pagina dikke MSX-Listingboek ter waarde van ƒ17,95 *
Stuur deze bon op naar: Wegener Tijl Tijdschriften Groep b.v.
Postbus 9943, 1006 AP Amsterdam

Reg. nr.

Naam: 4 1
T.a.v: 4 2
Straat: 2 1
Huisnr.: 2 6
Postcode: 2 8
Woonplaats: 2 2

MSX Computer Magazine verschijnt 8x per jaar
Een abonnement kost ƒ 50,–

Niet invullen
1 2
3 1
4 4

Met bestellen wacht ik op de acceptgiro, na betaling ontvang ik het gratis MSX-Listingboek

* Zo lang de voorraad strekt

MCM's Programma Service

Alle programma's uit dit nummer gebruiksklaar, met een redactionele extra! De makkelijkste en snelste manier om een eigen programma-bibliotheek op te bouwen. Dat bieden we u als extra service aan met onze Programma Service. Bespaar u de moeite van het intikken van lange listings (met alle risico's van fouten) en bestel alle programma's uit dit nummer, klaar om te draaien.
Let op: gironummer, adres en telefoonnummer van de Lezers-Service zijn gewijzigd!

Op cassette/diskette MCM/MSX-37 staan:

SCRSAV, onze utility die uw monitor beschermd tegen inbranden;
LIST, onze handige ML-hulp voor Basic-programmeurs;
ZAS-BAS, een dijk van een Z80-assembler. Onontbeerlijk voor de cursus machinetaal, met voorbeeldje;
Kort & Krachtig programmaatjes: STROFZ, ABHSTR, POTLOD, TEKST, CUJOMU en STORIN;
SCRPCH, de patch om AackoScribe aan te passen;
CMDEMO, een voorbeeld van een programma dat middels het ombuigen van een 'hook' extra Basic-functies creëert;
De hulpprogrammaatjes bij Tokens enzo: LINLEN en LINPRT;
Lezers Helpen Lezers met: DESPAT, DRDAT, KRKKLK en LC10DP. Sorry voor de namen, maar ja, ze moeten in zes tekens...
Uit de trukendoos: BRDCLR en TOBUF;
De patch voor DiskView uit de Oeps.
Voor de liefhebbers: wat voorbeelden van door MCBC – de MSX-compiler die in dit nummer besproken wordt – gecompileerde programma's.
Tenslotte: MCM's Invoer Controle Programma nummer 6.
Bovendien, alleen op diskette, schermen uit MCM's Art-Gallery!

Ook in de vorige nummers boden we u cassettes en diskettes met alle gepubliceerde programma's aan. Deze kunt u nog bestellen. Hieronder treft u een greep aan uit de beschikbare cassettes en diskettes.

MCM/MSX-33: Jungle, een adventure dat u voor raadsels zal stellen; Tijdmenu – en dit keer de goede versie van dit video-hulpprogramma dat timecodes op uw videotapes zet; FMPAK, ons eerste experiment met de nieuwe MSX-sound en wat Kort & Krachtig listinkjes. Bovendien, alleen op diskette, schermen uit MCM's Art-Gallery!

Met MCM/MSX-34 bieden we: Vuurwerk, zonder kruitdampen prachtige vuurpijlen op uw beeldscherm; Life, het bekende wiskundige spel in speciale MSX-uitvoering waarbij de cellen door vlammen zijn weergegeven, onder de naam Bos-Brand; Kerstkaart, een serie van drie MSX1 grafische hoogstandjes; WeWish, oftewel Kerstmuziek: We Wish You a Merry Christmas en nog eens drie andere kerstliedjes, waar we in het blad geen ruimte voor hadden.
Verder de Kort & Krachtig listinkjes, met onder meer een bloemlezing uit de haardvuren en uit de Lezers Helpen Lezers een patch voor de Canon T22A printer voor Ease. Bovendien, alleen op diskette, schermen uit MCM's Art-Gallery!

MCM/MSX-35 omvat: SAL90, salarisberekeningen voor het jaar 1990, inclusief de Oort-aanpassingen; Hola, een gokspelletje met een rad van avontuur; verder de Kort & Krachtig listinkjes: SuperPrint, waarmee u tekst groot kunt uitprinten, ook gekanteld; Pasmog, om passwoorden voor Maze Of Galious mee uit te rekenen; BRDR1 en BRDR2, smooth scrollen in de randkleur; RGNBG – wat staat voor regenboog – een hulpje bij het uitzoeken en selecteren van kleuren op scherm 8; en Wandel, laat letters van een titel één voor één het beeld in wandelen. Bovendien, alleen op diskette, schermen uit MCM's Art-Gallery!

MCM/MSX-36 bestaat uit: Fiscus90, uw aangifte makkelijk gemaakt; AxelF, machtige muziek voor het FM-PAC; PadKey, een programma om het Philips tekentableau als alternatief toetsenbord te gebruiken; uit de trukendoos: Breedte, Tstprt en Beeper; de vijf programma's uit het Achter de schermen van MSX artikel; het voorbeeld bij 'Gebruikers-vriendelijk programmeren' en de Lezers Helpen lezers listings Intlac5, Seppat2 en Effi. Tenslotte: MCM's Invoer Controle Programma nummer 6 en bovendien, alleen op diskette, schermen uit MCM's Art-Gallery!

PC-diskettes

Zolang de voorraad strekt zijn ook onze PC-diskettes nog verkrijgbaar. Zie vorige nummers voor een omschrijving.

Prijzen

De MCM-cassettes – alleen voor MSX – kosten, inclusief verzendkosten, *f* 12,50 per stuk. Diskettes kosten *f* 22,50 voor 3.5 inch en *f* 20,00 voor 5.25 inch, voor zowel MSX als PC, inclusief verzendkosten.
Lezers in België kunnen eveneens profiteren van de Programma Service. De prijzen in Belgische Francs: cassette Bfr. 250, diskette 3.5 Bfr. 450, diskette 5.25 Bfr. 400.

Hoe bestelt u?

1. Gireer het juiste bedrag (met vermelding van de juiste bestelcodes, zie de bon) naar postgironummer 6093761 t.n.v. MCM, Amsterdam. Uw overschrijving zegt ons precies wat we waarheen moeten zenden.

2. Of bestel schriftelijk: gebruik de bestelbon (kruis de juiste hokjes aan), en sluit een geldig betaalmiddel bij. Opsturen naar: MCM LezersService, postbus 75142, 1070 AC, Amsterdam.
U krijgt uw bestelling zo snel mogelijk thuisgestuurd.

Schrijf uw naam en adres – en uw bestelling – in duidelijke blokletters.

Voor vragen over de MCM Programma Service kunt u terecht op het telefoonnummer: 020-860743. De gewone vragenlijn van de redactie, waar u op het antwoordapparaat uw boodschap kwijt kunt. Wij bellen u zo spoedig mogelijk terug.

ONZE PROGRAMMA'S GEBRUIKSKLAAR OP DISKETTE OF CASSETTE

BON

– **MCM/MSX Cass. nrs.** ....
à *f* 12,50 / Bfr 250

– **MCM/MSX Disk 3.5 nrs** ....
à *f* 22,50 / Bfr 400

– **MCM/PC Disk 5.25 nrs** ....
à *f* 20,– / Bfr 400

– **MCM/PC Disk 3.5 nrs**..........
à *f* 22,50 / Bfr 400

Opsturen naar:

MSX Computer Magazine
Afdeling Lezersservice
Postbus 75142, 1070 AC, Amsterdam

Ja, ik maak gebruik van de Programma Service. Stuur de op deze bon aangegeven cassette(s)/diskette(s) naar het volgende adres:

Naam: ....

Adres: ....

Postcode: ....

Woonplaats: ....

Land: ....

S.V.P. invullen in BLOKLETTERS

Ik heb een geldig betaalmiddel bijgesloten (bijvoorbeeld een betaalkaart van bank of giro) en krijg mijn bestelling zo snel mogelijk toegestuurd. Alle prijzen zijn inclusief verzendkosten.

Nieuw elan

REDACTIONEEL



Misschien hou ik mezelf voor de gek, maar ik heb het gevoel dat MSX de laatste tijd weer lekker in de lift zit. Nee, geen nieuwe machines, althans geen Europese, maar wel een atmosfeer van 'we gaan er eens lekker tegenaan'. Kleinschalig, maar op zich prima.

Bijvoorbeeld op de Internationale Tilburgse Computerdagen. Ondanks de naam een puur MSX-gebeuren. De twee dagen dat we daar doorbrachten waren hectisch.
Om te beginnen, we hadden pas laat besloten aanwezig te zijn. Vervoer liet zich niet meer regelen, dus gingen we met twee personen – en een rugzak vol bladen en diskettes – op pad. Per trein, natuurlijk.
Dat leverde echter wel een vrij lege kraam op, die eerste zaterdag. Gelukkig was overbuurman Time Soft meteen bereid om in Amsterdam wat extra bladen op te pikken, die avond, zodat we met een gerust hart ons hotel konden opzoeken. Op zondag zag de MCM-kraam er heel wat gevulder uit...

De beurs zelf was een typisch geval van uit de jas groeien. Volgens de organisatoren waren wij niet de enigen, die zich pas laat aangemeld hadden. Vandaar dat er naast de twee zaaltjes een fikse feesttent was opgeslagen, om de laatkomers onder te brengen. Echt warm was het daar niet, overigens, tot de schrik van de standhouders daar. Die hielden de jas maar aan.
Tijdens die twee dagen heb ik nauwelijks achter mijn eigen kraam gebivakkeerd, hoewel ik nog een 'aardige' discussie met een software-piraatje heb mogen voeren. Iemand liet zich er op voorstaan slechts twee legale programma's te bezitten, maar op de vraag of hij zijn computer ook niet had kunnen jatten kwam geen antwoord.

Maar het leeuwedeel van mijn tijd heb ik nuttiger gevuld. Praten, praten en nog eens praten. Nuttige gesprekken met mensen die zich voor MSX inzetten, waar de komende nummers de nodige artikelen en besprekingen uit zullen voortvloeien. Dat, gekoppeld aan MCM's programmeursgroep – die nu van de grond begint te komen – maakt me toch optimistisch. Het is wat kleinschaliger dan pakweg drie jaar geleden, maar het bruist van de initiatieven.

Opvallend in Tilburg was – laat ik er eerlijk voor uitkomen – de rotherrie. Eén FM-PAC is mooi, een hele reeks die elkaar middels versterkers proberen af te troeven is echter wat al te veel van het goede. Geluid is in, bij MSX-minnend Nederland en België.
Net zoals het ombouwen van machines naar MSX2+. Ik heb zeker tien verschillende ombouw-projecten gezien. Wel iets trouwens om mee op te letten, aangezien de meeste van die ombouw-modellen niet helemaal foutloos functioneerden. Het datum-formaat bijvoorbeeld is vaak Japans, hetgeen een aantal programma's in verwarring zal brengen. Ook bij het PRINT USING commando zitten er nog wel wat addertjes onder het gras. Nog afgezien van de copyright-problemen – want ook een MSX2+ ROM mag niet zonder meer gekopieerd worden – is die ombouw blijkbaar niet zonder problemen. In een volgend nummer zullen we daar eens dieper op in gaan.

Al met al is het me uitstekend bevallen, die beurs. MSX Computer Magazine is wat al te lang niet actief geweest op de kleinere bijeenkomsten. Maar we zullen ons leven beteren, bijvoorbeeld op de door het CUC georganiseerde Nationale MSX & PC Computer Koop-beurs, op 9 juni. De locatie: de Groenoord-Hallen te Leiden.

Tenslotte, een heel andere zaak. De advertentie van HSH in de laatste nummers in MCM heeft heel wat stof doen opwaaien. Niet alleen was het in deze advertentie bedoelde – maar zorgvuldig niet bij name genoemde – bedrijf goed kwaad, ook diverse lezers hebben boze brieven gezonden. En dat kan ik me goed indenken, want echt zinnig is het natuurlijk niet om zo over straat te rollen.
Wat men zich echter niet zal beseffen is dat HSH zelf ook alles behalve gelukkig is met die advertentie. Tot twee keer toe heeft iemand binnen de uitgeverij nu kans gezien om de verkeerde advertentie op te nemen. Oftewel, deze als eenmalig bedoelde uiting is nu al twee keer verschenen in plaats van een gewone advertentie, waarin HSH de nieuwste MSX-snufjes wilde aanbieden. Dit keer neemt HSH geen risico's meer: men adverteert voor alle veiligheid gewoon niet.

Eerste Hulp Bij Overleven

Avonturiers, ruimtebonken, tijdreizigers en laser-kanonniers, wie vinden er nu geen baat bij MCM's Eerste Hulp Bij Overleven?
Geheime passwords, slimme trucs, zelfs POKE's om vals te spelen kunt U in deze rubriek vinden. Maar, om de E.H.B.O. te kunnen schrijven moet de redacteur van dienst wel uw hulp hebben.
Stuur in, die kaarten en verslagen van verre reizen door Computerland.

E.H.B.O. is het laatste – of eerste – toevluchtsoord voor diegenen onder ons die het ook allemaal niet meer weten. Bij E.H.B.O. kan men terecht voor informatie over adventure games, tips voor en over spellen én andere wetenswaardigheden.
Uw tips zijn altijd welkom onder het motto: Helpt Uw MSX Mede-computer-Mens!

Thunderbirds are go!

Is een bekende kreet uit de zestiger jaren. En dat de Thunderbirds nog steeds in de belangstelling staan blijkt wel uit de stapels post over dit programma.
Alleen, het is zo jammer dat bijna al die brieven doen vermoeden dat men of niet de handleiding leest of dat iedere briefschrijver over een illegale kopie beschikt. In elke brief wordt wel de klacht geuit dat om het spel te kunnen starten je een codewoord nodig hebt, maar dat men dat niet heeft.

Niets is echter minder waar! Thunderbirds bestaat uit vier, opeenvolgende delen. Alleen deel één is zo op te starten. Aan het eind van de eerste missie krijg je dan het codewoord om deel twee op te starten. Het uitspelen van deel twee geeft het codewoord voor deel drie enzovoorts. Passwords zouden voor Konami liefhebbers toch geen problemen mogen geven – en dat zijn we toch allemaal.
Nu heb ik in een vorige E.H.B.O beloofd de codes voor dit overigens voortreffelijke spel te geven, maar gezien het bovenstaande wil ik dit toch maar even uitstellen.

F1 Spirit

Is in een vorige E.H.B.O. uitvoerig besproken. Roland Noordermeer deed ons een uitvoerig schrijven toekomen waarin hij poogt de logica achter de F1 Spirit passwords te achterhalen. Wij citeren vrijelijk uit zijn brief.

Wanneer de passwords wat nauwkeuriger bekeken worden dat valt direct op dat bij één en dezelfde spelstand altijd 16 verschillende passwords van toepassing schijnen te zijn. Om te achterhalen hoe de passwords opgebouwd zijn moeten allereerst de passwords die bij een 'lege stand' – alles 0 – horen vergeleken worden. Bij een alfabetische rangschikking van de passwords blijkt dat de letters binnen een password altijd één letter opgeschoven zijn ten opzichte van het voorafgaande password. Tabel 1 laat zien hoe dit werkt.

Willen we een password ontleden, dan willen we in feite te weten komen welke plaats/cijfer binnen het password bij welke race hoort. Zie hiervoor tabel 2.

Belangrijk is uiteraard dat de standen gewijzigd kunnen worden. Roland heeft, zoals dat wetenschappelijk heet, op empirische wijze – veel uitproberen betekent dat gewoon – een tabel samengesteld waarin de verandering van een letter bij een bepaalde score is vastgelegd. De truuk staat in tabel 3.

Het is duidelijk dat het bovenstaande niet direct eenvoudig overkomt. En laat ik eerlijk wezen, ook ik heb hier een behoorlijke tijd naar zitten turen. Gelukkig gaf Roland een uitgebreid voorbeeld van hoe te werk te gaan.

Stel voor dat we een password willen hebben, dat begint met 'G' en waarin alles 0 is behalve F3000 en F1 Race ronde 4 (Monaco). Voor Monaco hebben we negen punten nodig – zie tabel. Als eerste moet dus een 'nul-password' beginnend met de letter 'G' samengesteld worden.

Tabel 1: Opbouw van de 'nul-passwords'

```
D E F G H I J K L M N O P
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
N D C D E C N M A A F K I B G C E F|A|P|C|A|K|K|B|C|N|I|M|C|M|A|N|
                                  | | | | | | | | | | | | | | | |
                                  A B C D E F G H I J K L M N O P
```

```
x.1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.18.19.20.21.x.x

x: begin letter A-P

1: rally
2: stock race
3: endurance race
4: F3 race
5: F3000 race

F1 races:

 6: round 1 : Brazil              14: round  9: Hungary
 7: round 2 : San Marino          15: round 10: Austria
 8: round 3 : Belgium             16: round 11: Italy
 9: round 4 : Monaco              17: round 12: Portugal
10: round 5 : USA                 18: round 13: Spain
11: round 6 : France              19: round 14: Mexico
12: round 7 : Great Britain       20: round 15: Japan
13: round 8 : West Germany        21: round 16: Australia

x.x : 2 letters voor de checksum
```

Tabel 2: Opbouw van een password

0	4	5	6	7	8	9
A	E	F	G	H	I	J
B	F	E	H	G	J	I
C	G	H	E	F	K	L
D	H	G	F	E	L	K
E	A	B	C	D	M	N
F	B	A	D	C	N	M
G	C	D	A	B	O	P
H	–	–	–	–	–	–
I	M	N	O	P	A	B
J	–	–	–	–	–	–
K	O	P	M	N	C	D
L	–	–	–	–	–	–
M	I	J	K	L	E	F
N	J	E	L	K	F	E
O	–	–	–	–	–	–
P	L	K	J	I	H	G

Tabel 3: Verandering van een letter

Doe dit als volgt: zoek in de bovenste rij van tabel 1 de 'G' op. Dit is de eerste letter van het password. De rest van het password wordt gevormd door alle letters op de middelste regel, te beginnen met de letter direct onder de 'G' op de bovenste regel. De laatste letter van het password is de letter op de middelste regel welke vóór het streepje van de letter G op de onderste regel staat. Dat klinkt gecompliceerd, maar reken het resultaat maar even terug:

G D E C N M A A F K I B G C E F A P C A K K

En nu wordt het pas echt ingewikkeld!

Om de punten te wijzigen van de F3000 race zoeken we in tabel 2 op op welke plaats de letter staat die bij F3000 race hoort. Dit is het cijfer 5, in het password in tabel 2 de zesde positie van links innemend.
Terugcheckend naar het door ons samengestelde 'nul-password' zien we dat op de zesde positie een 'M' staat.
In tabel 3 kijken we nu hoe de 'M' als 0-waarde verandert. Zoek in de 0-kolom de letter M op. Kijk vervolgens op dezelfde regel onder het aantal gewenste punten (=9) welke letter hier staat: 'F'. Dit betekent dat de 'M' op positie 6 gewijzigd moet worden in een 'F'.
Hiermee is de 0 van de F3000 race gewijzigd in een 9. Voor F1 race, ronde 4 doen we hetzelfde: de 'K' op de tiende positie van links moet gewijzigd worden in een 'D'.

Het uiteindelijke password ziet er dus als volgt uit:

G D E C N F A A F D I B G C E F A P C A K K x x

En daar loopt Roland met zijn MSX hoofd tegen de Konami muur. De beide x'jes aan het einde staan in de plaats van een controle checksum, waarvan Roland niet weet hoe deze berekend moet worden. Kortom, werk aan de winkel voor onze vaste password specialist Michel Shuqair uit Lelystad. En deze keer niet zoals met de berekeningen voor de passwords van Maze of Galious stiekem aan Kort & Krachtig opsturen, m'n waarde.

Andorogynus

Is een van die beruchte Japanse importen die onze joystickwiebelaar volledig uit z'n bol deden gaan – zestiger jaren term, betekent zoveel als extatisch raken tot bijna een religieuze ervaring aan toe. Wat gezien de spullen die in die tijd gerookt/geslikt werden niet verbazingwekkend moge wezen.

Ook ikzelf heb menig uurtje joystickwiebelend met Andorogynus doorgebracht. Jazeker beste lezer, af en toe kan zelfs ik toegeven 'den joystick te wiebelen'. Nu we een eindredactsel hebben dat probeert Kontra uit te spelen, moet dit kunnen.
Alhoewel, in de eindredactionele burelen wordt tegenwoordig over 'wizzen' gesproken in verband met het uiterst favoriete Quickshot joypadje dat we bijna allemaal gebruiken – ik heb het wat aangebrande exemplaar van de voorpagina van het vorige nummer geërfd.

Van een van mijn Belgische fans (hoop ik?), Jeffrey de Prêtre uit Zuienkerke kreeg ik een tamelijk verwijtend briefje dat hij het totaal oneens was met mijn gebazel over Andorogynus. Nu kan ik mij niet voorstellen dat ik ooit bazel – hoe het geacht eindhoofd daar ook over moge denken – en zeker niet over Andorogynus. Bij controle bleek al ras dat Jeffrey tegen enige uitlatingen van onze geachte spellenrecensent in de aanval ging. Onze wiebelaar schreef namelijk, en ik citeer hierbij letterlijk:

"Met voor Telenet ongebruikelijk goede MSX2 graphics, uitstekende muziek – één van de weinige muziekjes die ik vrolijk mee zit te fluiten terwijl ik speel; aanstekelijk – en goed-vloeiende bewegingen is Andorogynus een van de betere actiespellen. De 2 MB Mega ROM Andorogynus maakt duidelijk deel uit van een nieuwe generatie spellen.
Net als R Type is Andorogynus moeilijk, maar niet zo moeilijk dat beginnende spelers er niet aan hoeven te beginnen. Andorogynus bezit genoeg kwaliteiten om aangenaam bezig te houden. Het

spel is niet dusdanig snel en agressief dat je het al vrij spoedig opgeeft. In dat opzicht prefereer ik Andorogynus boven R Type. R Type is zo moeilijk dat alleen uiterst ervaren arcade actie fanaten eraan moeten beginnen. Voor Andorogynus moet je wel een stevige portie doorzettingsvermogen hebben, maar inzicht is net zo hard nodig als snel reageren."

Tot zover het citaat van mijn geachte collega, wat Jeffrey in de pen deed klimmen. Tot op heden heeft hij slechts twee maal de tweede grot van Andorogynus weten uit te spelen. Hij wijst nadrukkelijk op zijn ervaring en stelt nadrukkelijk dat we niet te maken hebben met een 'verwaaide joystick-rukker'. Een lange lijst van uitgespeelde spellen is bijgesloten.

Nu is het normaliter niet zo dat de wiebelaar en ondergetekende op dezelfde golflengte zitten, maar – jammer voor Jeffrey – beiden hebben we in Andorogynus menig grotje betreden. Nu kan het natuurlijk zijn dat onze expertise beter is – alhoewel Jeffrey's opsomming van spellen indrukwekkend is, Nemesis 3 in één dag – of dat het gebruik van het 'wizje' tot betere resultaten leidt (neen, Belgische bieren komen in dit blad niet ter sprake).

Hoe het ook zij, Andorogynus blijft een grandioos spel dat zeker uit te spelen is! En het argument van Jeffrey dat honderd gulden voor een spel dat ontzettend moeilijk is, teveel is kan ik niet onderschrijven. Je krijgt in zo'n geval zeker waar voor je geld; anderzijds, hoe voel je je als je honderd gulden uitgegeven hebt en je speelt het spel binnen het uur uit??

Ondanks zijn negatief begin was Jeffrey vol lof over onze joysticker en bood direct aan hem te helpen bij het schrijven van zijn recensies. Wat die lof dan waard is?

R-Type

Daarnaast had Jeffrey enkele tips voor R-Type. Met excuses voor de vaktaal.
De gekleurde blokjes die men – na het afschieten van een bepaald soort lopend vehikel met een bolle kop – kan oppikken bevatten extra wapens, zoals 'botsende lasers' en 'kurketrekker-lasers'. Ook de wapenpod verschijnt pas na het oppikken van zo'n gekleurd blokje. De rode bolletjes met een grijze mantel vol-gen je gelijk de opties uit de verschillende Nemesis'en; ze schieten echter niet maar dienen meer als schild.
Het Boss monster aan het eind van level één zal velen nachtmerries bezorgen. Ga als volgt te werk:

Bij het opkomen van de Boss schiet je alle ogen van hun steeltjes. Zoals je misschien gezien hebt bevindt zich halverwege de buik van het monster een grijze streep met kwijl...
Vlieg naar die streep als de Boss zijn staart omhoog steekt. Let op! Raak die *staart* niet! Uit de grijze streep zie je opeens een kopje omhoog steken. Wijk naar achteren, laadt je laserkanon tot het maximale vermogen en vlieg wanneer de staart omlaag is, op het kopje af en blaas het in elementaire deeltjes.

In level twee beperk je je bij de vijand tot het kapot schieten van de segmenten tussen de twee koppen en het ontwijken van vijandelijke kogels. Na verloop van tijd – stug volhouden – geraak je in level drie.

In dat derde niveau moet het grote schip langs de onderkant benaderd worden. Gebruik de pod om de kanonnen aan de onderzijde van het grote schip onschadelijk te maken. Vlieg door tot het einde van het schip. Schiet her en der nog wat hinderlijke elementen uit het grandioze decor en ga dan over tot het alledaagse werk: totale vernietiging van het grote schip.
Hang de pod achter je en desinfecteer de bovenzijde van het schip. Laadt het energiekanon van je schip maximaal op en geef het harmonica een terminale stoot energie op zijn blaasbalg. Level vier zal nu je deel zijn.

En daar laat ik het even bij. Na het bovenstaande zal nog meer R-Type jargon wel eens fataal voor het voortbestaan van de E.H.B.O. kunnen blijken. De hoek waaruit het gevaar dreigt moge overduidelijk zijn. Meer tips zijn overigens welkom. Iemand al geheime velden in R-Type ontdekt?

Poke,s

Deze stromen nog steeds niet binnen. Veel kan ik deze keer dus niet bieden. Wat losse flarden.

Na de passwords voor Wizard's Lair in de vorige aflevering nu wat poke's om het einde uit te stellen. Laadt het eerste gedeelte van de cassette; dit is in Basic. Verwijder in de laatste regel het RUN commando en voeg de volgende regels toe:

```
1000 POKE &H9178,l
1010 POKE &H9177,g
1020 POKE &H9172,s
1030 POKE &H9173,r
1040 POKE &H9174,d
1050 SCREEN2
1060 DEFUSR =
PEEK(64703) + 256*PEEK(64704)
1070 X = USR(0)
```

Hierbij geldt:

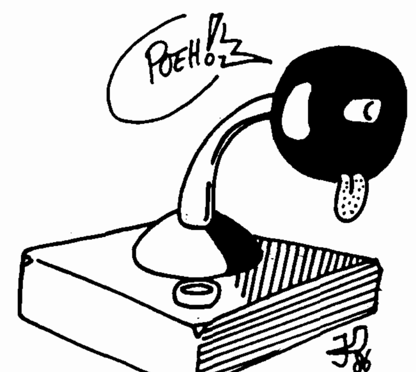
l = aantal levens;
g = startvoorraad goud;
s = aantal sleutels;
r = aantal ringen;
d = aantal diamanten.

Let op! Er zijn in de loop der tijden twee loaders voor dit spel gebruikt. Bij de oudste loader kun je bovenstaande regelnummers gebruiken. Bij de nieuwere loader moet je even uitkijken dat je geen bestaande regelnummers overschrijft.

Starquake

Starquake is evenals Wizard's Lair een programma van het reeds geruime tijd ter ziele zijnde Engelse software huis Bubble Bus - herinnert iemand zich dat steeds crashende biljart programma Hustler? Om de hoeveelheid levens fors uit te breiden pas je de volgende truc toe.
List de loader en verwijder het RUN commando uit de laatste regel.
Voeg de volgende regels toe:

```
1000 POKE &H9135,x
1010 SCREEN2
1020DEFUSR = PEEK(64703) + 256*
PEEK(64704)
1030 X = USR(0)
```



SCRSAV: spaart uw monitor

Wie niet zo rijk is dat hij of zij een kleurenmonitor kan betalen, zal meestal met een monochroom scherm werken. Deze kijkkastjes hebben wel als nadeel dat een letter – of een heel scherm – langer zichtbaar blijft dan de bedoeling is. Ook als de buis uitstaat: de zaak kan domweg inbranden. Voorkom dit probleem en verleng de levensduur van uw monitor!

EEN HULPJE VAN KLASSE

Het verschijnsel dat letters blijven nagloeien op een monitor zullen de meeste mensen wel kennen. Wat heet, de fosfor in de monitor – de stof die oplicht als er een electronenstraal op valt – is daar speciaal op samengesteld. Anders kregen we collectief schele hoofdpijn van het flikkeren van het beeld.
Toch is dat nagloeien soms hinderlijk, vooral als twee opeenvolgende schermbeelden sterk op elkaar lijken. De 'oude' informatie blijft nog even zichtbaar, met alle leesproblemen van dien. Soms is een kwartier later nog te zien dat er een mooie grafiek op het scherm stond!

Te lang

Dat nagloeien komt vooral voor als het scherm een tijd lang statisch is gebleven, bijvoorbeeld als men even aan de telefoon zit onder het werk. Inbranden is de term die hiervoor gebruikt wordt. Meestal wordt het beschouwd als een hinderlijk effect, dat maar op de koop toe moet worden genomen.
Wat veel mensen echter niet weten, is dat door dat inbranden het beeldscherm een vreemde voorkeur krijgt voor de plaatsen waar al vaker informatie lange tijd heeft staan pruttelen. Anders gezegd, op die plaatsen zal een volgende keer het beeld weer iets langer zichtbaar zijn. En u raadt het al: dat gaat zo door. Plekken die eenmaal ingebrand zijn worden steeds gevoeliger.
Een ander hinderlijk effect – dat gelukkig pas na langdurig inbranden optreedt – is dat op een gegeven ogenblik de oplichtende laag aan de binnenkant van de beeldbuis opbrandt. Het fosfor raakt als het ware uitgeput. Nieuwe informatie zal dan op die plaatsen helemaal niet meer zichtbaar zijn – of ooit nog worden. MCM zou MCM niet zijn als we niet zouden proberen om de lezers te helpen met hun problemen...

De oplossing

De beste oplossing om uw monitor te sparen, is nog steeds de simpelste: gewoon niet aanzetten dat ding. Dat dit niet makkelijk typen is, mag duidelijk zijn. De gedachtengang achter de oplossing die wel werkt – en die wij dus gekozen hebben – is als volgt. Als de computer niet in gebruik is, maar wel aanstaat, zal de gebruiker al enige tijd geen toets meer ingedrukt hebben. Schakel dan het beeldscherm uit en wacht totdat de computer weer gebruikt wordt. Oftewel, als er een toets wordt ingedrukt, schakel het beeldscherm dan weer in en begin opnieuw met aftellen...

De techniek

Gelukkig hebben de ontwerpers van de videochip voorzien in de mogelijkheid om het beeldscherm uit – en weer aan – te zetten. Dat gebeurd door in de registers van de videochip één bit om te schakelen. Een techniek die gewoonlijk voor heel andere doeleinden wordt toegepast, want door het uitzetten van het beeldscherm wordt de videochip wat sneller. Maar dat is voor deze toepassing niet belangrijk. Wel is het van belang te weten hoe we dat bit heen en weer kunnen zetten.
Vanuit Basic gaat het uitzetten met:

```
VDP(1) = VDP(1) AND &HBF
```

Weer aanzetten doet men met:

```
VDP(1) = VDP(1) OR &H40
```

In machinetaal is dit indirect natuurlijk ook mogelijk, maar in de ROM zitten al routines ingebakken die deze functies vervullen. Op adres 41H vinden we DISSCR en op 44H zit ENASCR, die achtereenvolgens het scherm uit- en aanzetten. Er zijn geen parameters nodig, dus zouden deze routines ook vanuit Basic met een:

```
DEFUSR = &Hxx: A = USR(0)
```

gedaan kunnen worden, waarbij xx staat voor de waarde 41 of 44.
Maar met aan- en uitzetten alleen zijn we er niet. De belangrijkste eigenschap van het programma wat we willen hebben is dat het scherm pas na een tijdje wachten wordt afgeschakeld. Een wachttijd die we zelf willen kunnen bepalen. Hiervoor kan de hook H.TIMI gebruikt worden, die 50 keer per seconde aangeroepen wordt. De MSX-interrupt, met andere woorden. De huishoudelijke routine die bijvoorbeeld ons toetsenbord in de gaten houdt en bijhoudt welke toetsen er zoal worden ingedrukt. Als we aan die hook nu eens een simpel tellertje ophangen, dat na

een bepaalde tijd het scherm uitzet. Een tellertje dat steeds weer – als er een toets wordt ingedrukt – op nul gezet wordt... En dat is nou net de grap, met een MSX. Er zijn handige hooks voor bijna alles wat men zou kunnen willen. Zoals de hook H.KEYC, die telkens aangeroepen wordt als er een toets ingedrukt is. Door deze zaken te combineren krijgen we een geheel dat wacht totdat de computer enige tijd niet gebruikt is en dan pas het beeldscherm uitschakelt.

Het programma

De hierbij afgedrukte listing van SCRSAV gebruikt alle hierboven genoemde truukjes en nog meer...
Om te beginnen, SCRSAV kent een viertal verschillende manieren om zichzelf in uw MSX te nestelen. Al naar gelang uw werkwijze kan het programma zich op verschillende manieren manifesteren.Wie het intikt en opstart – eerst wegschrijven natuurlijk – zal een vijftal mogelijkheden tegenkomen. Keuze 5 is om de wachttijd van het programma te veranderen van de standaard drie minuten naar elk gewenst aantal seconden, tot 21 minuten.
Met de keuzes 1 en 2 zetten we SCRSAV gewoon in het geheugen, maar wel op verschillende plaatsen. Wie onder Basic blijft werken, kan SCRSAV het beste in de top van het geheugen installeren middels keuze 2. Wie naar DOS zou kunnen gaan, kan SCRSAV het beste in de PLAY buffer installeren met keuze 1, mits het zeker is dat er geen PLAY-commando's gebruikt zullen worden!
Deze laatste mogelijkheid is ook verwerkt onder keuzemogelijkheid 4, maar dan wordt SCRSAV als SCRSAV.BIN weggeschreven naar disk, zodat de Basic loader niet meer nodig is. BLOAD "SCRSAV.BIN",R is dan genoeg. Jammer genoeg bestaat deze handige mogelijkheid niet voor de andere Basic-variant, waarbij het programma in de Basic-geheugenruimte geplaatst wordt. Dat is alleen maar mogelijk vanuit de Basic-lader zelf.
Voor wie eigenlijk nooit met Basic opstart, maar altijd onder DOS werkt, is optie 3. Hiermee wordt een .COM file aangemaakt op de diskette, die door een AUTOEXEC.BAT file kan worden opgestart en die SCRSAV in de playbuffer neerzet. SCRSAV mag absoluut niet twee keer achter elkaar geladen worden, dan gaat de computer onvoorspelbare dingen doen!

Geheugen-gegoochel

Eigen roem heeft een geurtje, maar we moeten eerlijk zijn, het Basic gedeelte van SCRSAV zit doordacht in elkaar. De Basic laadt de machinecode – die in de data-regels staat – in het geheugen. En hoewel als allereerste actie het programma geheugen bovenin het Basic-gebied reserveert, wordt die ML in de PLAY-wachtrij gezet.
Die machinetaal bestaat op zijn beurt trouwens ook uit twee stukken, een installatie-routine en het eigenlijke programma, dat uiteindelijk het werk voor ons opknapt. Als we er voor gekozen hadden SCRSAV in het Basic-geheugen onder te brengen wordt dat stuk uit de PLAY-wachtrij naar die reeds gereserveerde Basic-bytes overgebracht, waarna de zaak wordt doorgestart via de installatie-routine. Het resultaat: de PLAY-wachtrij is weer geheel ter beschikking, terwijl het geheugenbeslag op de kostbare Basic-ruimte minimaal blijft.
Als we het hele programma in de PLAY-queue willen houden, dan zal de initialisatie de eigenlijke code toch wat opschuiven. Om precies te zijn naar de derde van de drie PLAY-wachtrijen. Wie wil kan vervolgens, met SCRSAV op de achtergrond in die derde wachtrij toch nog gebruik maken van de eerste twee queue's. Oftewel, toch nog tweestemmige muziek. En die reeds gereserveerde Basic-ruimte, die wordt weer braaf vrijgegeven.

Wie een .BIN bestand aanmaakt, die doet in feite iets wat sterk op het voorgaande lijkt. SCRSAV wordt – inclusief de initialisatie – in de PLAY-wachtrij geplaatst, maar niet opgestart. Dat stukje geheugen wordt vervolgens weggeschreven naar diskette – cassette, voor diegenen die geen drive bezitten.
Met een BLOAD "SCRSAV.BIN",R pikt men dan daarna de draad weer op: de initialisatie verschuift de werkende code naar de derde PLAY-wachtrij en het feest kan beginnen.

Getruukt

Helemaal getruukt is de wijze waarop SCRSAV voor MSX-DOS wordt voorbereidt in een .COM bestand. Eerst zet de Basic het eigenlijke programma weer in de PLAY-buffer, waarna het in het juiste formaat wordt weggeschreven – natuurlijk dit keer alleen naar disk – als SCRSAV.COM. Maar wel met een lichtelijk aangepaste initialisatie in de machinetaal. Dit .COM bestand wordt vervolgens, als het onder DOS als commando geladen wordt, vanaf 100 hex in het geheugen geplaatst. De kenners slaan nu natuurlijk alarm, als ze gezien hebben dat dit .COM programma met BSAVE is aangemaakt. Want hoe kan zo'n BSAVE-geval – waar in de eerste zeven bytes allerlei adres-gegevens staan – nu in hemelsnaam onder MSX-DOS worden gestart.
De oplossing is simpel, althans volgens de programmeur. Die opent na de BSAVE het bestand weer als random file en overschrijft die eerste zeven bytes met NOP'jes, de ML-code die niets doet. Het resultaat: laden vanaf 100 hex, starten op 100 hex en dan eerst zeven bytes lang niets doen voordat de actie start.
Die actie is natuurlijk weer de initialisatie: omhoog gooien van de eigenlijke code naar de derde muziekwachtrij. Waarna MSX-DOS weer keurig gereed staat voor gebruik. Met SCRSAV in de achtergrond.

De praktijk

Als SCRSAV eenmaal draait en er wordt geen toets aangeraakt, zal na verloop van tijd het scherm 'wegvallen'. Het scherm blijft weg – en de computer stilstaan! – totdat er een willekeurige toets wordt ingedrukt. Als er iets anders dan ESCape werd ingedrukt, zal het hele proces weer van voren af aan beginnen. Als het echter wel ESCape was, zal SCRSAV zichzelf uit het geheugen verwijderen, en dus ook niet meer terugkomen totdat het opnieuw geladen wordt. Bij een relatief korte wachttijd en een langdurige bezigheid van de computer – waarbij geen toetsen aangeraakt hoeven te worden – zal dus het beeldscherm vroeg of laat wegvallen. Na één druk op ESCape zal het niet weer gebeuren...

Tot slot

Veel succes met SCRSAV – en nu maar hopen dat niemand per ongeluk z'n computer uitzet als hij hem wil aanzetten, omdat hij denkt dat die uitstaat omdat z'n monitor nog aanstaat, maar geen beeld meer vertoont...
Alleen, dit begeleidende artikel. Voordat we dit product van onze nieuwe redactie-programmeur, Paul te Bokkel, eenmaal goed omschreven hadden. Het programma was natuurlijk geen enkel probleem, daar staat Paul wel garant voor. Ook zijn tekst – de basis voor dit

```
10 REM SCRSAV loader                           0
20 REM                                         0
30 REM MSX Computer Magazine                   0
40 REM                                         0
50 REM by PtB                                  0
60 REM                                         0
70 REM SCRSAV SLECHTS 1 KEER LADEN!            0
80 REM                                         0
90 CLEAR 200,PEEK(&HFC4A)+256*PEEK(&H
FC4B)-94                                     180
100 SCREEN 0: WIDTH 37: KEY OFF: T=90
00 ' 3 minuten **********************        110
110 IF PEEK(&HFD9F)=&HC3 AND PEEK(&HF
DCC)=&HC3 THEN PRINT "Al geladen!": E
ND                                           142
120 LOCATE 0,3: PRINT "Installeren Sc
reensaver": LOCATE 0,6: PRINT "[1] in
 play buffer": LOCATE 0,8: PRINT "[2]
 in top van geheugen"                        252
130 LOCATE 0,10: PRINT "[3] als A:SCR
SAV.COM": LOCATE 0,12: PRINT "[4] als
 SCRSAV.BIN": LOCATE 0,14                    104
140 PRINT USING "[5] verander wachtti
jd (##.# min)";T/3000: LOCATE 5,17: P
RINT "Geef uw keuze [1..5] ? ";               50
150 A=VAL(INPUT$(1)): IF A<1 OR A>5 T
HEN GOTO 150 ELSE PRINT CHR$(8);A            190
160 B=&HF975: LD=B: C=&HFA75+2^16: ON
 A GOTO 260,180,200,260,220                  254
170 ' adres voor residente gedeelte v
eranderen voor highmem versie *******          0
180 C=PEEK(&HFC4A)+256*PEEK(&HFC4B):
GOTO 260                                      12
190 ' laad adres van installatie rout
ine aanpassen voor .COM file ********          0
200 LD=&H107-2^16: GOTO 260                   14
210 ' aanpassen van wachttijd *******          0
220 CLS: LOCATE 0,3: PRINT "Geef nieu
we wachttijd (sec) ";: X=POS(0)               15
230 LOCATE X,3: INPUT T$: IF VAL(T$)=
0 OR VAL(T$)>1310 THEN GOTO 230              114
240 T=50*VAL(T$): CLS: GOTO 120              207
250 ' niets anders voor playbuffer en
 BLOAD file, of al veranderd ********          0
260 FOR AD=B TO B+145                         25
270  READ A$: IF ASC(A$)<>42 THEN PO
KE AD,VAL("&H"+A$): GOTO 290                 210
280  ON VAL(RIGHT$(A$,1)) GOSUB 410,
430,450                                       72
290 NEXT                                     224
300 ' geheugenversies starten *******          0
310 DEFUSR=B: IF A<3 THEN B=USR(0): O
N A GOTO 380,390 ELSE IF A=4 THEN GOT
O 360                                        111
320 ' save .COM, schoon 7 bytes *****          0
330 BSAVE "a:scrsav.com",&HF975,&HFA0
7: OPEN "A:scrsav.com" AS 1 LEN=7            172
340 FIELD #1,7 AS A$: LSET A$=STRING$
(7,CHR$(0)): PUT 1: CLOSE: GOTO 380           85
350 ' wegzetten .BIN file ***********          0
360 BSAVE "scrsav.bin",&HF975,&HFA07,
&HF975                                       211
370 ' vrijgeven geheugen voor alles b
ehalve highmem versie ***************          0
380 CLEAR 200,PEEK(&HFC4A)+256*PEEK(&
HFC4B)+94                                    172
390 END                                      190
400 ' tijd invullen *****************          0
410 POKE AD-1,T-INT(T/256)*256: POKE
AD,T/256: RETURN                             211
420 ' wegzetadres installeerroutine i
nvullen ******************************         0
430 V=PEEK(AD-1): V=V+LD+2^16: POKE A
D-1,V-INT(V/256)*256: POKE AD,V/256:
RETURN                                       188
440 ' runadres residente gedeelte inv
ullen ********************************         0
450 V=PEEK(AD-1): V=V+C: POKE AD-1,V-
INT(V/256)*256: POKE AD,V/256: RETURN         98
460 ' machinecode *******************          0
470 DATA 11,00,*3,21,34,*2,01,5E,00          150
480 DATA ED,B0,F3,21,9F,FD,11,47,*3          103
490 DATA 01,05,00,C5,ED,B0,C1,21,CC           12
500 DATA FD,ED,B0,3E,C3,32,9F,FD,32          103
510 DATA CC,FD,21,00,*3,22,A0,FD,21           95
520 DATA 54,*3,22,CD,FD,FB,C9,2A,53          128
530 DATA *3,2C,28,41,2A,51,*3,23,22           11
540 DATA 51,*3,11,00,*1,E7,20,34,2F          198
550 DATA 32,53,*3,CD,41,00,FB,76,F3           23
560 DATA 2A,FA,F3,3A,F8,F3,BD,28,F4          220
570 DATA 22,F8,F3,7E,D6,1B,20,12,21          130
580 DATA 47,*3,11,9F,FD,01,05,00,C5           72
590 DATA ED,B0,C1,11,CC,FD,ED,B0,CD          106
600 DATA 44,00,AF,32,53,*3,00,00,00          122
610 DATA 00,00,50,74,42,00,00,00,00          129
620 DATA 00,E5,21,00,00,22,51,*3,E1            2
630 DATA 18,EE                               163
```

artikel – was best doordacht. Maar zodra het echt spannend werd en de precieze truuks binnen de Basic-lader en de ML-initialisatie routine omschreven moesten worden, liet hij het lafhartig afweten. Een korte kreet over drie 'slimme' adressen, dat was alles. We hebben hem ondervraagd, tot het naadje van deze kous bekend was. Want anders zouden die dapperen die de precieze werking willen doorgronden van een zeer koude en uiterst frustrerende kermis thuiskomen. Volgende keer graag iets meer uitleg, Paul!

EMCIEBICIE, MSX-Compiler?

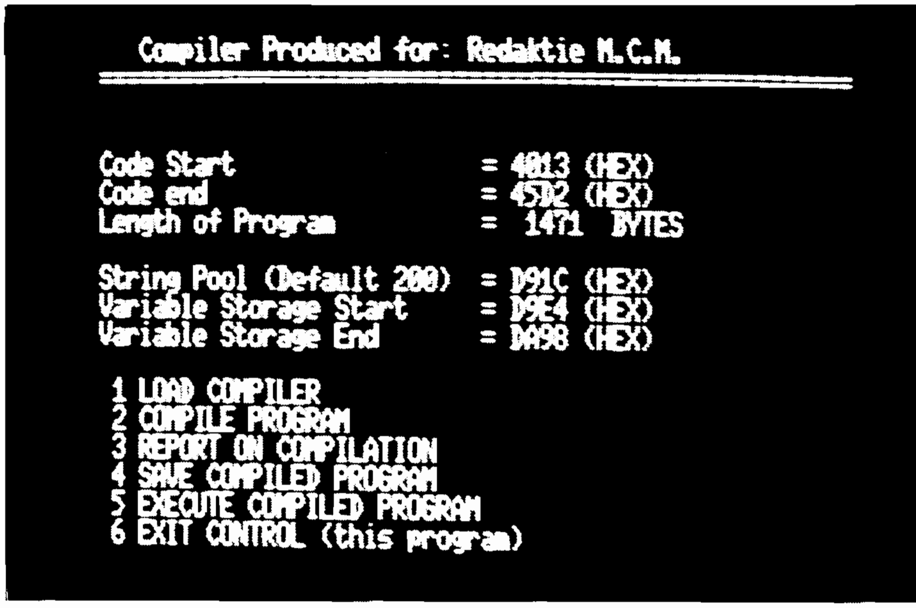
Wat er voor de MSX nog steeds niet is en eigenlijk al lang had moeten zijn is een compiler. De Belgische MSX Club heeft een poging gedaan dat te veranderen. Het resultaat draagt de cryptische naam MCBC, een naam die door de meeste voorbijgangers op zijn Engels wordt uitgesproken.

Een MSX compiler op het bureau van de hoofdredacteur blijft daar natuurlijk niet lang liggen. Het barst van de redacteuren die naar wel eens naar willen kijken. Meestal komt het er op neer dat er verschillende mensen een korte blik werpen en er uiteindelijk één persoon het ding echt uitspit.
Dat uitspitten is natuurlijk leuk werk als een programma of stuk hardware goed werkt. Er zijn echter ook gevallen dat het niet alles goud is wat er blinkt. Om te weten tot welke categorie MCBC behoort hoeft u alleen maar door te lezen.
De ervaringen van een programmeur die ooit – vanwege de rampzalige traagheid van de ZX81 – machinetaal leerde en daar tot op de dag van vandaag geen spijt van heeft.

Eerste indruk

Een goede eerste test van een programma is meestal het gewoon eens te starten. Bij MCBC was dat wat lastig, omdat er dan toch eerst een te vertalen stukje Basic moet zijn. Maar daar komen we zo nog op terug.
Die eerste rondwandeling bleef – bij gebrek aan een te vertalen programma – beperkt tot het starten van de demonstraties op de disk, die inderdaad een redelijk indrukwekkende snelheid te zien gaven. Het bleek mogelijk met behulp van gecompileerde Basic enige tientallen sprites vloeiend over het scherm te bewegen. Iets dat normaal gesproken op een MSX niet lukt.
Ook het voorbeeld waarin het scherm volgebouwd wordt met steeds iets verdraaide H-vormen was indrukwekkend. Tijdens de gewone Basic-versie smaakte de koffie uitstekend. De gecompileerde versie bood zelfs te weinig tijd een nieuwe bak in te schenken.
Kort en goed: de voorbeelden waarin veel gerekend wordt met kleine getallen, zoals schermcoördinaten, hebben veel baat bij het compileren.
De snelheidswinst wordt minder als er 'zwaardere' instructies gebruikt worden. Zeker de opdrachten die door de MSX videochip uitgevoerd kunnen worden – voornamelijk de lijn- en blokinstructies op scherm 4 en hoger – bieden weinig mogelijkheden tot versnelling.
Of het lijncommando nu door Basic of door een machinetaalprogramma naar de VDP gestuurd wordt maakt niet veel uit. De videochip voert in beide gevallen het werk op dezelfde – behoorlijk hoge – snelheid uit.

DE NIEUWE MSX-COMPILER GETEST



Verslag van de compilatie en het hoofdmenu van CONTROL

Praktijk

Maar goed, een eerste blik in de handleiding leverde een redelijk beeld op van wat er wel en niet mee gecompileerd kon worden. En toen was het tijd voor een eigen probeersel.
Het programmaatje TEST1.BAS bevatte een blok tekst van vier regels dat opgeslagen was in een string-array. Enkele FOR-NEXT constructies zorgde ervoor dat de tekst van links af het scherm in schoof en vervolgens ook weer naar links verdween. Te traag in Basic, ideaal voor compilatie.
Maar MCBC werkte niet mee. Vreemde tekens op het scherm, vele beeps en zelfs een reset waren het gevolg. Lang spitten – programma opdelen en stukje voor stukje opnieuw compileren – leverde geen zinnig antwoord op. Pas toen iemand op het idee kwam eens een:

```
PRINT A$(3)
```

in te voegen bleek wat er fout ging: string-array's worden niet (goed) ondersteund. En dat terwijl de handleiding letterlijk zegt dat "één dimensionale array's zowel van tekst als integertype" ondersteund worden. Het probleem – of eigenlijk het programma – is uit de wereld geholpen met een KILL.
Tijdens deze experimenten ontstond de behoefte even een nieuwe disk te formatteren. De compiler – die aangeroepen wordt met CALL COMP – maakte dat echter onmogelijk. Het CALL FORMAT commando eindigde in een Syntax error. Gelukkig lukte het formatteren na uit en weer aanschakelen wel...

Handleiding

Diezelfde handleiding bevat trouwens nog enige slordigheidjes. Zo wordt er ergens de functie 'VPEEK(a,b)' beschreven. Daarnaast worden zoals in

MSX-land wel vaker gebeurd de termen pagina en blok slordig gebruikt. Er wordt op een gegeven moment zelfs gesproken over pagina 'vier'. Hopelijk wordt daarmee blok vier van de memory mapper bedoeld, want elke MSX deelt de 64K adresruimte op in vier pagina's, genummerd van nul tot en met drie!
Het typisch Vlaamse taalgebruik en het regelmatig noemen van de naam van de schrijver van MCBC – ene Adriaan – maken het geheel wel redelijk leesbaar. Het stijltje is bijna familiair, alsof de schrijver het naast je zit uit te leggen. Wat dacht u van 'Slierten zonder duidelijk begin en eind dat lust een compiler niet'. Die zin zou ook in een Italiaans kookboek kunnen staan! Het boekwerkje is behoorlijk volledig. Er is een lijst van ondersteunde commando's, compleet met korte uitleg en mogelijke 'omwegen'. Ook worden de niet ondersteunde commando's opgesomd.
Achterin is zelfs een lijst opgenomen met mogelijke problemen en de oplossingen daarvoor. Voor de echte beginner wordt er ook een inleiding gegeven over wat een compiler eigenlijk is. Kortom: de opzet is goed, de afwerking kon beter. Door het hele verhaal heen wordt verwezen naar het MSX Club Magazine. Niet alleen wordt de indruk gewekt dat het een leuk blad is om te lezen, maar tevens wordt er gemeld dat men het zal moeten lezen voor meer gegevens over MCBC. Op basis van de handleiding raden we iedereen aan een abonnement te nemen, voor zover men dat nog niet heeft.

Subset

MCBC ondersteunt niet de volledige MSX instructieset. Op zich is dat niet vreemd. Er zitten namelijk in elk Basic-dialect instructies die eigenlijk meer met het besturingssysteem en de programma-ontwikkeling te maken hebben dan met iets anders. Geen enkele MSX compiler hoeft zaken als LIST, LLIST, AUTO, TRON, TROFF, SAVE, LOAD, CSAVE, CLOAD? en dergelijke te ondersteunen.
Het wordt al twijfelachtig bij zaken als FILES, LFILES, KILL, NAME en dergelijke. Ook dat zijn in principe functies die in het besturingssysteem thuishoren en niet in de taal, maar het is in programma's toch vaak makkelijk dat ze er zijn. Eén van de keuzes die de ontwerpers van de compiler hebben gemaakt is het achterwege laten van alle diskfuncties. Het lezen en schrijven van bestanden is – ook uit machinetaal – traag, zo luidt de redenatie. Op zich waar, maar dit heeft wel onmiddellijk tot gevolg dat programma's die gegevens naar disk schrijven of van disk lezen slechts ten dele gecompileerd kunnen worden.
En voor u zich nu grenzeloos verbaast, MCBC is inderdaad in staat om een Basic-programma slechts ten dele te compileren. Juist om die functies die de compiler niet ondersteunt alsnog in een programma te kunnen gebruiken.
Verder heeft men voor MCBC de keuze gemaakt alleen met integers te werken. Het hele floating point gebeuren is overboord gegooid.
Op zich is dat ook niet vreemd. Er is op dat terrein niet veel snelheid te winnen en het zou de compiler een stuk complexer maken. Toch zou het mooi zijn als er ooit eens een compiler zou verschijnen die dergelijke grappen wel kan uithalen.
Deze twee keuzes maken ook commando's als FIELD, GET, PUT, OPEN, LOF(), SIN, COS, ATN, SQR en dergelijke overbodig. Het grootste deel van de lijsten niet ondersteunde- en niet zinnig om te ondersteunen commando's is hiermee gevuld. Er staan echter ook instructies tussen die wij wel graag in een compiler zouden willen zien. LOCATE bijvoorbeeld is een doodeenvoudig commando dat in vrijwel elk programma voorkomt. Kennelijk vinden veel programmeurs het niet handig de cursor aan het begin van de negende regel te zetten met:

```
PRINT CHR$(27);"Y( ";
```

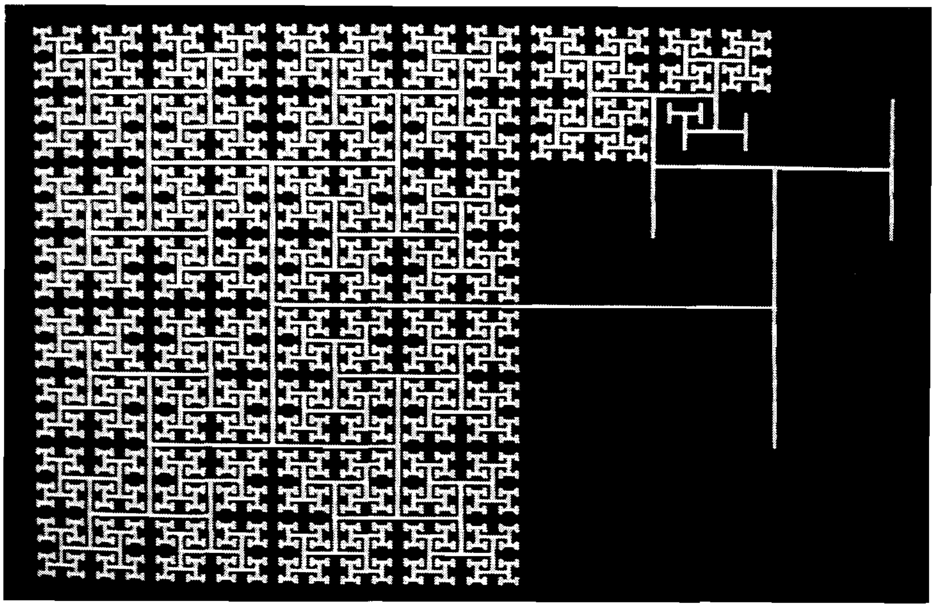
zoals men in de handleiding suggereert. Maar het kan natuurlijk zijn dat u daar anders over denkt.
SWAP is een soortgelijk geval. Het kan zo mogelijk nog eenvoudiger dan een LOCATE herschreven worden, maar het is in feite een doodsimpel commando. Dat moet een compiler gewoon kunnen. Hetzelfde geldt overigens voor DATA. Leuk om te zien: DATA wordt niet ondersteund en daardoor vind men RESTORE niet zinnig om te ondersteunen. Dit riekt naar een cirkelredenering. Maar inderdaad, men kan het DATA-commando ontlopen door alle variabelen rechtstreeks van een waarde te voorzien, middels een is-gelijk teken. Alleen, of het handig is... Op dezelfde manier ontbreken ook TAB, SPC, SPACE$,

Niet ondersteunde commando's

ATN	DSKO$	LOC	RSET
BASE	EOF	LOCATE	SIN
BEEP	ERASE	LOF	SPACE$
BIN$	EXP	LOG	SPC
CIRCLE	FIELD	LSET	SQR
CLOSE	FN	OPEN	STRING$
COS	FPOS	PLAY	TAB
CSRLIN	GET	POS	TAN
DATA	HEX$	READ	USING
DRAWDSKF	INPUT	RESUME	VAL
DSKI$	INSTR	RND	WIDTH

Niet zinnig om te ondersteunen

ATTR$	CVD	LIST	OCT$
AUTO	CVI	LLIST	RESTORE
BLOAD	CVS	LPOS	RENUM
BSAVE	DELETE	LPRINT	RUN
CALL	END	LOAD	SAVE
CDBL	ERLERR	MAXFILES	SWAP
CINT	ERROR	MERGE	TROFF
CLOAD	FILES	MKD$	
CMD	FRE	MKI$	
CONT	IPL	MKS$	
CSAVE	KILL	MOTOR	
CSNG	LFILES	NAME	



Een fraaie demo

BEEP – ja, er is een BIOS routine, maar toch – CSRLIN en WIDTH. Allemaal commando's die maar net ingewikkelder zijn dan REM. Niet zinnig geacht worden dingen als LPRINT en LPOS. Hoe is het toch mogelijk dat wij vooral de eerste zo vaak gebruikt hebben? Al met al is de lijst instructies die MCBC wel ondersteund nog behoorlijk groot, maar toch vinden wij het overdreven om te stellen dat MCBC MSX Basic kan compileren. De kans dat een willekeurig Basic programma direct door de compiler gehaald kan worden is erg klein. In vrijwel alle gevallen zal er het één en ander verbouwd moeten worden. Onze pogingen om bestaande MCM-programmatuur 'even' te compileren hebben we ietwat gefrustreerd gestaakt.

Werking

Het compileren zelf gaat vrij rap. Er is gekozen voor compileren uit RAM naar RAM, wat het proces natuurlijk erg snel maakt. Er staat op de schijf een programmaatje CONTROL waaruit het geheel te besturen is. De eerste optie in dit CONTROL-menu is het inladen van de compiler. Zoals te verwachten is heeft dat een positief effect op de rest van de resultaten.

De compiler zelf blijft na een reset in het geheugen aanwezig. Vandaar dat deze optie in het menu zit. Het zou natuurlijk al mooier zijn als CONTROL even zou controleren of de compiler al aanwezig is. Nu moet de gebruiker dat zelf bijhouden.

Optie 2 is het compileren. De gebruiker komt na het kiezen van deze optie terug in Basic, met de opdracht het te compileren programma te laden, een CALL COMP commando te geven en vervolgens CONTROL weer te starten.

Dan biedt CONTROL de mogelijkheid de gegevens van de compilatie te bekijken, het gecompileerde programma te bewaren en de aangemaakte machinetaal te starten.

Voor de echte freaks is er gelukkig nog een andere manier om MCBC te gebruiken, het kan ook helemaal zonder CONTROL. De compiler zelf bevindt zich namelijk in blok 2 van de memory mapper, terwijl de gecompileerde Basic in blok 3 en eventueel 4 terecht komt. Door nu zelf de mapper aan te sturen is het mogelijk blok 2 in pagina 1 te schakelen en de compiler te starten met 'CALL COMP' of blok 3 in pagina 1 te schakelen en de machinetaal te starten met 'CALL MEM'.

Even een verklaring voor de hier gebruikte terminologie, die verschilt van de handleiding van MCBC maar gelijk is aan wat de technische documentatie zegt. De mapper is verdeeld in *blokken* van 16Kb, de nummering begint bij 0. In dat blok staan in principe de hooks, systeemvariabelen, disk-Basic met de buffers en de stack. De 64Kb adresseerruimte is verdeeld in *pagina's* van 16Kb. Pagina 0 beslaat de adressen &h000 tot en met &h3FFF, terwijl pagina 4 van &hC000 tot en met &hFFFF loopt.

De compiler maakt een stuk machinetaal dat kan lopen op adres &h4000. Wanneer het juiste mapper blok is ingeschakeld kan de ML gestart worden met een CALL MEM instructie. Het is natuurlijk mogelijk meerdere pagina's van de mapper te vullen met verschillende ML routines. Op die manier kunnen uit een (Basic) hoofdprogramma verschillende snelle stukken ML worden aangeroepen, hetgeen nu precies de manier is waarop MCBC een Basic-programma in meerdere te compileren delen kan splitsen.

Meer experimenten

Natuurlijk is een eerste indruk niet voldoende. Veel belangrijker is een tweede poging. Na de handleiding dus bestudeerd te hebben probeerden we het nog eens.

Om te beginnen een programma dat 2 tot de macht honderd uitrekent. Dus 2x2x2x2x... x2, honderd tweetjes en 99 kruisjes. Het antwoord is natuurlijk gigantisch groot. Het past nooit in een integer, zelfs niet in de 14-cijferige MSX floating point variabelen. Vandaar dat we het antwoord opslaan in een array van 80 elementen, met in ieder element precies één cijfer. Het geheel is nu eenvoudig – maar traag – met twee te vermenigvuldigen door cijfer voor cijfer te vermenigvuldigen en eventueel eens iets te onthouden. Precies de methode die op de basisschool ook toegepast wordt.

In Basic werkte het prima. Na enkele koppen koffie stond inderdaad de honderdste macht van twee op het scherm. Tot vlak na het compileren bleef alles naar wens.

De eerste reactie toen we de ML zagen lopen was hoopvol. Razendsnel ratelde de teller omhoog. Perfect. Maar bij de 40 was het om onduidelijke redenen uit met de pret. De tweede versie – inderdaad, per ongeluk een verkeerd commando ingetikt – kwam bijna twee keer zo ver, maar alweer brak het rekenen voortijdig af met rotzooi op het scherm.

Een derde probeerseltje gaf ook puinhoop. Het idee was een lijntje over het scherm te laten stuiteren, dat een spoor achterlaat. Voor de gelegenheid is de programmering maar eens niet al te snel gemaakt, snelheidsverhogende circulaire buffers zijn simpelweg achterwege gelaten.

Ook hier weer: in Basic werkt het perfect, maar de gecompileerde versie laat het afweten. De lijntjes worden niet goed gewist, maar veranderen meestal slechts van kleur. Zelfs het veranderen van de XOR in het bot tekenen van een zwarte lijn mag niet baten. Even Apeldoorn bellen?

Wat is een compiler?

Kort en goed is een compiler een programma dat een programmatekst omzet in machinetaal die rechtstreeks door de computer uitgevoerd kan worden. Verreweg de meeste hogere programmeertalen zijn ontworpen om gecompileerd te worden, Basic is de uitzondering die deze regel bevestigd.
Compileren heeft namelijk een groot nadeel. De complete programmatekst moet aan de compiler gevoerd worden, die dan aan het vertalen slaat. Meestal zal de programmatekst wel ergens een foutje bevatten, dus volgt er een lijst foutmeldingen.
Dan begint het gedonder. De editor – een apart programma – moet gestart worden en de tekst geladen. Pas dan kan de fout verbeterd worden en de tekst weer weggeschreven. Dat alles in de hoop dat de compiler alles nu wel slikt.
Maar als een programma eenmaal succesvol gecompileerd is zijn we er nog niet. Tijdens de uitvoering van het programma kunnen er nog foutsituaties optreden. Bekende voorbeelden zijn delen door nul of het te groot of te klein worden van variabelen. Dergelijke fouten heten 'run-time' errors. Ze zijn meestal moeilijker te vinden en te verbeteren dan de fouten die de compiler meldt.
Om de nadelen van het compileren te omzeilen en studenten sneller met de basisprincipes vertrouwd te maken is ooit Basic ontworpen. Deze taal word al sinds het allereerste begin geïnterpreteerd. Dat wil zeggen dat een interpreter de instructies stuk voor stuk bekijkt en op basis daarvan bepaalde acties uitvoert. Deze manier van werken heeft als nadeel dat de uitvoering van het programma traag wordt. Behalve aan het werkelijk uitvoeren van de instructies is de computer immers ook veel tijd kwijt aan het interpreteren ervan. En die twee klussen wisselen elkaar continu af. Het voordeel echter is dat de tijd tussen editten en testen nihil is. In Basic R, U, N en return.
Omdat Basic een relatief eenvoudig te leren taal is, werden vrijwel alle huiscomputers van een interpreter voorzien. Dat maakte Basic behalve makkelijk te leren ook erg bekend. Een zeer groot deel van de mensheid heeft zijn of haar eerste programmeerervaring in Basic opgedaan. En dat is iets dat door programmeurs die hun sporen intussen verdiend hebben betreurd wordt.

Wat dan wel?

MCBC wordt verkocht als MSX compiler. Welnu, een MSX compiler in de zin dat er MSX Basic in gaat en machinetaal uit komt is het absoluut niet. Het enige dat dit programma met MSX te maken heeft is dat het er op draait.
Er zitten vrij veel eigenaardigheden en zelfs enkel bugs in de door ons geteste versie. Wie er mee aan het werk gaat betreedt een hobbelig pad. Maar waarschijnlijk zullen de problemen door de ondersteuning van MSX Club Magazine wel opgelost worden – op den duur.
Wie een spel wil programmeren, geen machinetaal kent en toch snelheid wil kan met MCBC veel bereiken. De demo's spreken boekdelen en ook onze eigen experimenten – ze komen zowel in Basic als gecompileerd op de bij dit nummer behorende diskette – waren snel. Voor zover ze werkten, althans.
Kortom: wie bereid is speciaal voor de compiler te gaan schrijven en zich aan te passen aan het programma wensen we veel plezier met MCBC. Maar iedereen die denkt zijn zelfgeschreven boekhoudpakketje even door de compiler te kunnen halen komt van een (ijs)koude kermis thuis. Het kan, maar is een hels karwei.
Al met al, we hebben het gevoel dat MCBC een goed initiatief is, dat echter nog allesbehalve uitontwikkeld is. Zoals het nu staat is het een alternatieve taal voor MSX, die inderdaad qua snelheid prima resultaten weet te behalen. Maar een compiler in de zin waarin dat woord meestal gebruikt wordt is het in onze ogen niet. De goede documentatie doet daar niets aan toe of af.

MCBC
Prijs: *f* 75,-
Verdere informatie:
MSX Club België-Nederland
p/a Mottaart 20
B-3170 Herselt
België

Liever geen Basic

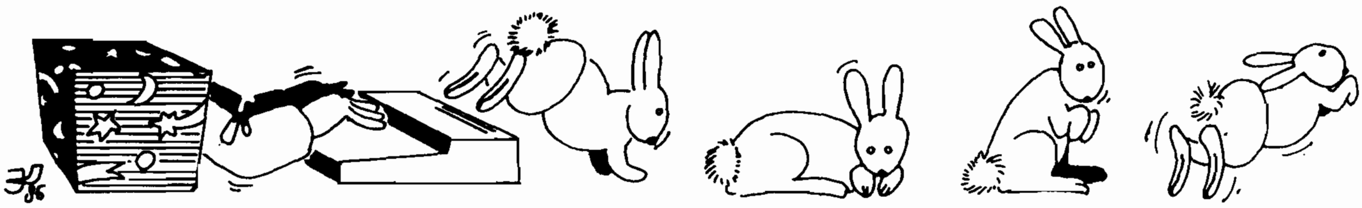
Omdat Basic niet voor compilatie is ontworpen is het compileren van een Basic programma erg lastig. Het bouwen van compilers is op zich al een belangrijk onderdeel van een studie informatica, maar Basic is beduidend moeilijker dan de meeste andere talen.
Om te beginnen is Basic ambigu. Dat wil zeggen dat een Basic-regel meerdere betekenissen kan hebben. Een belangrijk voorbeeld zijn dingen als:

```
IF A>B THEN IF C>D THEN INST1 ELSE INST2
```

In de situatie dat A < B is het maar de vraag of INST2 uitgevoerd moet worden. Maar het wordt de compiler al lastig gemaakt met een opdracht als: PRINT A

want wat is die A? Een integer, een dubbele precisie variabele of een gewone floating point? Dat lijkt misschien niet zo belangrijk, maar als er een getal op het scherm gezet moet toch echt eerst duidelijk zijn om wat voor type het gaat. Een interpreter kan dat wel uitzoeken als dat nodig is, maar een compiler moet dat direct kunnen zien. Soms geeft een DEFINT commando uitkomst, maar in MSX Basic kan dezelfde A in dezelfde regel voor verschillende types staan.
Het allergrootste probleem is namelijk dat de zogenaamde 'program flow', ofwel de volgorde waarin de instructies uitgevoerd worden, zeer onduidelijk kan worden. Vrijwel alle andere talen kennen gestructureerde methodes om lussen en keuzes te implementeren. Basic doet het voornamelijk met GOTO. De enige taal die het op dit punt van Basic 'wint' is machinetaal. Een ander voorbeeld is de instructie CLS. Die werkt anders, afhankelijk van het schermtype dat op het ogenblik van uitvoering actief is. Deze dubbelzinnigheid is voor de interpreter geen probleem – het herkennen van acht verschillende soorten CLS is even lastig als het herkennen van acht verschillende schermtypes – maar compilers kunnen er moeilijk mee overweg. Er zijn nog veel meer instructies waarvoor geldt dat de interpreter ook de toestand van de machine en de eventuele argumenten betrekt in de beslissing wat er eigenlijk gebeuren moet. Een compiler moet een beslissing nemen voordat er ook maar één instructie uitgevoerd is. Het is dan nagenoeg onmogelijk de toestand van de machine en de waarden van de argumenten te voorspellen.

De trukendoos



Computers zijn behoorlijk complexe apparaten. Telkens weer blijken er dingen te kunnen die niemand ooit voor mogelijk had gehouden. Vaak ook blijkt een probleem een stuk eenvoudiger op te lossen dan het lijkt. Handleidingen zouden al dergelijke truuks eigenlijk moeten vermelden. Ze doen – en laten we eerlijk zijn: kunnen – dat echter niet. Vandaar de Trukendoos, een rubriek waar het oneigenlijk gebruik van commando's en instructies een hoofdrol speelt. Hoewel we ook niet vies zijn van het 'getruukt' in de luren leggen van een programma

ONMISBAAR VOOR DE WARE LIEFHEBBER

In de vorige aflevering zijn er nogal wat POKE's behandeld. Deze keer zullen we eens gaan kijken naar de leuke DEFUSR's. Het BIOS van elke MSX machine bevat namelijk een aantal routines die in principe bedoeld zijn om uit machinetaalprogramma's aangeroepen te worden.
Meestal is er vanuit Basic weinig mee te doen, omdat er gegevens in bepaalde registers gezet moeten worden. Maar wanneer er niet met de routine gecommuniceerd hoeft te worden is een stukje BIOS ook vanuit Basic makkelijk bruikbaar. En sommige routines kunnen zeer nuttig zijn!
Om uit Basic een machinetaalroutine te kunnen starten moet het beginadres bekend zijn. Dat adres kan dan gedefinieerd worden als het beginadres van één van de tien mogelijke machinetaal-routines. In Basic worden ze aangeduid met USR0 tot en met USR9. Het is mogelijk het cijfer achter de USR weg te laten, Basic gaat er dan van uit dat USR0 bedoeld wordt.
Om een beginadres toe te wijzen aan één van de tien USR's is er de DEFUSR instructie. Een ML routine op adres &hC000 kan bijvoorbeeld gedefinieerd worden met:

```
DEF USR0 = &hC000
```

Vanaf dat moment kan USR0 gebruikt worden als elke andere Basic-functie. Het is zelfs mogelijk een getal of een string aan de ML routine mee te geven en net als elke functie levert USR ook een waarde op. Voorbeelden van het gebruik zijn:

```
PRINT USR0 (12)
A = USR1 (0) + USR2 (0)
B$ = USR5 ("Dit is een string-argument")
```

In het bovenstaande voorbeeld zou USR5 bijvoorbeeld een routine kunnen zijn om alle kleine letters in een string om te zetten in hoofdletter. USR1 en USR2 leveren kennelijk getallen op, terwijl USR0 duidelijk een bewerking op 12 loslaat.
Het gaat nu echter om BIOS routines die geen invoer en geen uitvoer hebben. Tussen de haakjes vullen we dan meestal een 0 in, er moet immers iets staan. Het resultaat wordt opgevangen in een variabele die verder niet gebruikt wordt. Er ontstaat dan een constructie als:

```
X = USR0 (0) of
DUMMY = USR(0)
```

Wat er dus moet gebeuren om één van de onderstaande routines aan het werk te zetten is:

```
DEF USR = < adres >
DUMMY = USR(0)
```

Genoeg uitgelegd nu. Eerst wat BIOS-routines die uit Basic aangeroepen kunnen worden en daarnaast nog nuttig zijn ook. Daarna stappen we over naar de andere truuks.

54) Reset (0)

Een beetje flauw is de BIOS 'routine' op adres 0. In feite begint daar de initialisatie van de computer zoals die plaatsvindt na het resetten of inschakelen van de MSX.
Na het aanroepen van deze USR keert de controle dan ook niet terug naar het aanroepende programma, maar volgt er een reset. In sommige gevallen kan het makkelijk zijn dit uit een programma te kunnen forceren.

55) Reset F-toetsen (&h3E)

Functietoetsen niet meer in orde? Niet Apeldoorn bellen maar simpelweg even &h3E aanroepen. Eenvoudig en doeltreffend: de originele inhoud van de functietoetsen is in no-time weer hersteld.

```
10 REM BRDCLR: Goochelen met de randkleur                                    0
20 REM                                                                       0
30 REM Ingezonden door Michel Shuqair, Nijmegen                              0
40 REM                                                                       0
50 REM MSX Computer Magazine                                                 0
60 REM                                                                       0
70 SCREEN 0: WIDTH 80: COLOR 15,4,1: DEFINT P: VDP(13)=PEEK(&HF3EA)+PEEK(&HF3E9)
*16: COLOR ,PEEK(&HF3EB): FOR P=&H800 TO &H90D: VPOKE P,255: NEXT P: VDP(14)=16  170
```

Alleen op het scherm blijft de oude inhoud nog even staan. Door even op een shift toets te drukken word het scherm ook bijgesteld aan de werkelijke inhoud van de toetsen.

Om uit het programma ook het scherm even aan te passen aan de functietoetsen kan natuurlijk ook even de BIOS routine op adres &hC9 worden aangeroepen. Als de functietoetsen inderdaad op het scherm staan zal die routine het scherm en het geheugen weer op één lijn brengen.

56) Scherm uit (&h41) en aan (&h44)

De VDP kent de mogelijkheid het scherm uit te schakelen. Het hele beeld wordt dan gevuld met de achtergrondkleur. Op die manier heeft de VDP geen tijd meer nodig om het beeld op het scherm te zetten en kan de communicatie met de rest van de computer sneller verlopen.
Door adres &h41 aan te roepen krijgt het hele scherm de achtergrondkleur. De normale – werkbare – situatie kan weer hersteld worden door een USR naar adres &h44.

57) Sprites wissen (&h69)

Soms blijven er lastige sprites op het scherm achter. Die kunnen eenvoudig gewist worden door deze routine. Na een aanroep is er geen sprite meer te zien. Wat valt er verder nog te zeggen?

58) Geluid uit (&h90)

De PSG kan op verschillende manieren aangestuurd worden. Direct via SOUND instructies of via het muzikale PLAY commando. Vooral wanneer de laatste manier gekozen wordt is het niet eenvoudig de muziek netjes af te breken. In Basic althans...
Want in machinetaal – en door het ontbreken van in- en uitvoer ook in Basic – kan BIOS routine &h90 aangeroepen worden. Die zorgt ervoor dat alle spelende muziek keurig stilgelegd wordt. Heel wat netter dan het BEEP-commando, wat hier vaak voor misbruikt wordt.

59) Buffer leeg (&h156)

Een ingedrukte toets wordt niet direct aan het lopende programma doorgegeven. Eerst plaatst het BIOS de toets in een buffer waarna het programma de code uit de buffer leest. Op die manier is het mogelijk snelheidsverschillen tussen het programma en de gebruiker soepel op te vangen.
Toetsdrukken die niet direct verwerkt kunnen worden blijven tijdelijk in de buffer bewaard.

Soms werkt het systeem echter niet zoals de bedoeling is. Er zijn momenten dat er codes in de buffer blijken te zwerven waar niemand meer op zit te wachten. Om dan te voorkomen dat die toetsen herkend worden moet de buffer gewist worden.

Een veel gebruikte mogelijkheid om dat voor elkaar te krijgen is het aanroepen van de BIOS routine op adres &h156. Er is echter nog een alternatief, een methode die keurig in Basic werkt:

```
10 IF INKEY$< >"" THEN GOTO 10
```

Deze regel haalt een toets uit de buffer (INKEY$) en als die er inderdaad was gebeurt het nog eens. Pas als er geen toets meer in de buffer zit – en INKEY$ dus een lege string teruggeeft – wordt de lus verlaten en een eventuele volgende instructie uitgevoerd.
Een ander alternatief dat eigenlijk hetzelfde doet als de BIOS routine is:

```
POKE &hF3FA,PEEK(&hF3F8)
POKE &hF3FB,PEEK(&hF3F9)
```

Deze POKE instructies zorgen ervoor dat de systeemvariabelen die naar het begin en het eind van de buffer wijzen gelijk worden. Daardoor ontstaat een situatie die door het BIOS herkend wordt als een lege buffer.

60) Border in 0

Eén van de eerste Basic-commando's die MSX'ers leren is COLOR. Zoals bekend kan de schermkleur ermee ingesteld worden. Voor de meeste schermen kunnen er drie getallen ingevuld worden: één voor de kleur van de tekst, één voor de achtergrondkleur en tenslotte één voor de kleur van de rand.
Die rand – ook wel bekend als 'border' – kan niet gebruikt worden om tekst of tekeningen in te zetten, de computer kan er simpelweg niet komen. Het enige dat kan is het ding een kleurtje geven. Dat wil zeggen: op de meeste schermen. In screen 0 is zelfs dat niet mogelijk. Tenzij u toevallig een MSX2 bezit en in 80-kolommen mode werkt.

In dat geval zit er in de VDP namelijk toch de mogelijkheid verborgen de rand een ander kleurtje te geven. Alleen wordt dit natuurlijk niet in de handleiding vermeld en werkt het COLOR commando niet.
Om de rand een eigen kleurtje te geven moet er wat met VDP's en VPOKE's gegoocheld worden. Het 'programma' BrdClr van Michel Shuqair doet dat. Een VDP'tje hier, een VPOKE'je daar en klaar is kees. Alle kleuren staan keurig zoals dat door de COLOR in de listing bevolen werd.

Dit programma is een typisch voorbeeld van wat er in de trukendoos thuishoort. Iedereen die de werking ervan doorgrondt zal dat beamen. De truuk die gebruikt is, is eigenlijk gewoon flauw, maar het werkt perfect!

61) POKE -1

Sinds de invoering van de MSX2 bestaat de bekende 'POKE -1' instructie, die dient om software die anders niet op de twee zou lopen toch aan de gang te krij-

```
10 REM TOBUF: Toetsbuffer genept          0
20 REM                                    0
30 REM MSX Computer Magazine              0
40 REM                                    0
50 A$="? "+CHR$(34)+"Dit is MCM's TOB
UF"+CHR$(34)+CHR$(13)+"? "+CHR$(34)+"
Nog eens..."+CHR$(34)+CHR$(13)          151
60 FOR F=1 TO LEN(A$): POKE &HFBEF+F,
ASC(MID$(A$,F,1)): NEXT F               151
70 POKE &HF3FA,240: POKE &HF3FB,251     216
80 POKE &HF3F8,(LEN(A$)+240)MOD256: P
OKE &HF3F9,INT((LEN(A$)+64497!)/256)    121
```

gen. De twee meest bekende varianten zijn:

POKE -1,170 en
POKE -1,255

Laten we hier over één ding vooral duidelijk zijn: programma's die volgens de standaard geschreven zijn werken zowel mèt als zonder extra POKE instructie perfect.
Alleen de software die niet goed met de regels omgaat moet op deze manier de weg gewezen worden.

Het probleem is dat de POKE voor elk type computer anders is. Daardoor is het lastig de instructie in programma's op te nemen, want die programma's kunnen dan alleen op dat type computer werken. Het is dus makkelijker gebruik te maken van de volgende variant:

POKE -1,(NOT(PEEK-1))AND240)
*17/16

Dit zal op elke MSX2 werken, omdat de instructie zelf uitzoekt welke waarde er gePOKE'd moet worden. De uitdrukking achter de komma levert op elk type computer de goede waarde op. Met deze regel in een programma hoeft niemand zich meer zorgen te maken over adres -1 of eigenlijk: 65535.

62) Turbo machinetaal

Speciaal voor ML-programmeurs: bij de MSX wordt nogal eens een blok geheugen naar een I/O poort geschreven. Meestal zal aan de andere kant van de poort de VDP staan om de reeks bytes op te vangen. Standaard kent de Z80 een instructie die voor een dergelijke actie geknipt is:

OTIR, opcode: ED B0

Meestal is de snelheid echter van groot belang en er bestaat ook hier een truuk om het sneller te doen. Er moeten dan echter wel wat offers gebracht worden. Een OTIR instructie kost 21 klokcycli per byte. Alleen als teller B de waarde 0 bereikt heeft zijn er slechts 16 cycli nodig. De volgende serie instructies verplaatst twee bytes uit het geheugen naar een poort en kost 34 cycli, dat zijn er 17 per byte:

```
POP DE      (10 cycli)
OUT (C),D   (12 cycli)
OUT (C),E   (12 cycli)
```

Natuurlijk doet OTIR wat meer, zo wordt er ook een teller bijgehouden. Zo'n teller zal er dus nog bijgeplaatst moeten worden, bijvoorbeeld door het gebruik van een DJNZ instructie. Dat zou echter de tijdwinst weer teniet doen...

Voor dit probleem bestaat echter een standaardoplossing: herhalingen kunnen geprogrammeerd worden door botweg de instructies vaker in het programma te zetten. Door bovenstaande serie acht keer te herhalen en er vervolgens een DJNZ omheen te zetten blijft de tijd per byte binnen de 18 cycli. Ten opzichte van de OTIR is dat een winst van meer dan 3/21 ofwel zo'n 15 procent. Niet slecht.
Er zijn natuurlijk nadelen. Om te beginnen worden er per doorgang van de lus ineens 16 bytes verplaatst. Dat maakt dat er alleen veelvouden van 16 bytes verplaatst kunnen worden.
Ook moet er op gelet worden dat de bytes nu niet van oplopende maar van aflopende adressen gehaald worden. De serie simuleert eerder een OTDR dan een OTIR instructie. Uiteindelijk maakt het allemaal echter weinig uit, tellen is tellen.

Daarnaast word SP gebruikt. Dat heeft tot gevolg dat de interrupts uitgeschakeld moeten worden. De adressen die er bij de aanroep van de interrupt routine op de stack geplaatst worden zouden anders onze data vernielen. Verder moet SP natuurlijk gered worden. Na deze snelle routine moeten we natuurlijk weer door kunnen werken alsof er niets gebeurd is.
Moraal van het verhaal: het kan (bijna) altijd sneller. Denk niet te snel dat een oplossing perfect is, want door op het ene terrein wat in te leveren valt er aan een andere zijde vaak nog wat te winnen. Overigens: probeer deze versnelling maar niet op een MSX1, de VDP in die machine kan het tempo niet aan. In alle latere versies is het op deze manier wel mogelijk snel een (groot) blok gegevens naar het videogeheugen te 'blazen'.

63) Toetsbuffer genept

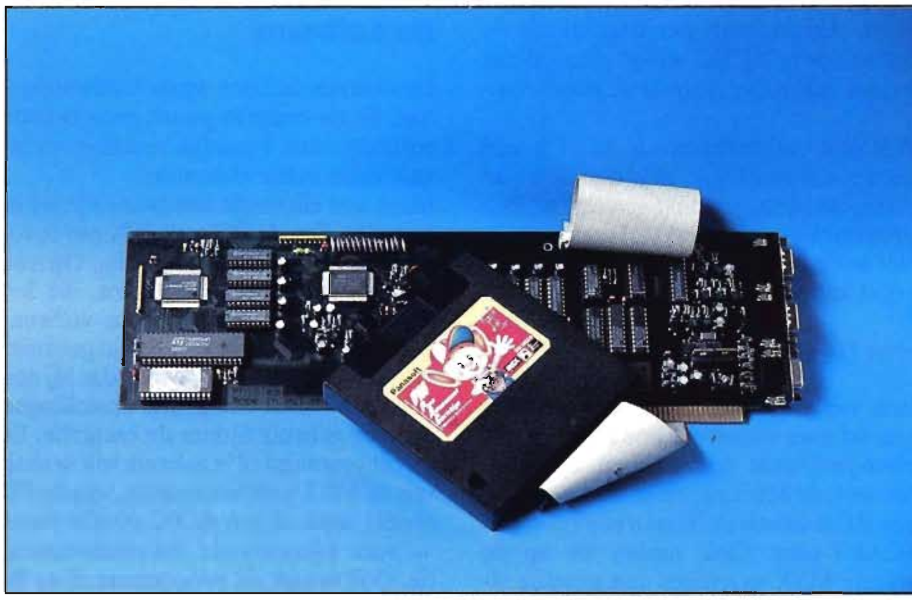
De toetsenbordbuffer is er om de invoer van het toetsenbord tijdelijk op te slaan. Maar het is natuurlijk mogelijk de zaak te neppen. Niemand verbiedt het een programma een teken in de buffer te zetten, in plaats van ze er uit te halen.
Deze techniek biedt mogelijkheden. MCM-programma's als Joysor en Padkey maken er gebruik van. Maar ook in Basic krijgt de programmeur er ineens mogelijkheden bij.

Het geven van een LIST, CLOAD of DELETE instructie bijvoorbeeld werkt wel uit een programma, maar de uitvoering ligt na die instructie wel stil. Door een dergelijke instructie in de toetsenbuffer te plaatsen met een return-code – CHR$(13) – erachter wordt hij ook uitgevoerd, maar nu is het mogelijk ook een GOTO of iets dergelijks in de buffer te plaatsen om het programma weer op gang te helpen.
Het programma ToBuf zet A$ in de buffer. Als demonstratie staan er in die string twee print-instructies, gescheiden door een returncode. Elk van de print-instructies kan natuurlijk vervangen worden door welke Basic instructie dan ook. Enige voorwaarde is dat A$ niet langer wordt dan 39 tekens, want dan past het niet meer in de buffer. Hier wordt door het programma niet op getest, een te lange A$ kan zeer vreemde gevolgen hebben....

Tot zo ver deze trukendoos. Volgende keer gaan we verder, hopelijk met een flinke serie ingezonden truuks. Er is nog ruimte op de plank, geachte lezers en lezeressen. Al uw bijdragen voor deze rubriek zijn welkom!

Een MSX in uw PC: de MSX-insteekkaart

Al zeker drie jaar geleden was het dat ons de eerste geruchten over een MSX als PC-insteekkaart bereikten. Een snel telefoontje leerde ons toen dat Philips daar inderdaad aan werkte, maar liefst nog geen ruchtbaarheid gaf aan dit idee. Sindsdien is er wel het één en ander gebeurd op de MSX-markt. Philips heeft de MSX aan de wilgen gehangen, zodat ook de laatste Nederlandse MSX-leverancier van het toneel verdween. Het is dan ook lang geleden dat MSX Computer Magazine een nieuwe MSX kon bespreken, de Japanse importen daargelaten.



En de MSX-kaart, die stak zo nu en dan wel de kop op in de bladen, maar Philips zou dit prima idee zeker niet meer uitbrengen. De geheugenprijs was één van de redenen die deze kaart commercieel niet haalbaar maakten... Echter, de prijzen van RAM op de wereldmarkt zijn ondertussen weer een heel stuk gedaald. En de hardnekkige verhalen dat die MSX-insteekkaart toch nog zou verschijnen zijn ondertussen waar gebleken. De PTC – de Philips Thuiscomputer Club – zit nu eenmaal dicht bij het vuur, zo blijkt maar weer.

Het idee

Het idee was simpel: een insteekkaart voor een willekeurige MS-DOS machine waarop een complete MSX zou zijn ondergebracht. Die MSX zou dan zoveel mogelijk gebruik maken van de hardware van de PC waar die kaart in geplaatst zou worden. Met dat simpele basisidee viel er ook al heel wat te speculeren. Zoals: wat voor monitor gebruikt die inbouw-MSX. Of: kan de eventuele harddisk ook gebruikt worden. En bovendien, hoe zouden ze dat met de slots oplossen. Zullen er cartridge-poorten zijn, en hoeveel dan wel. Oftewel, toen we die PTC-kaart onlangs eindelijk onder ogen kregen was dat een soort bekroning van drie jaar discussies op de redactie.

LANG GEWACHT, STIL GEZWEGEN...

De praktijk

Voor de PC is de insteekkaart wat bij een MSX een cartridge is: een stukje hardware op een eigen printkaart. Maar waar bij de MSX die kaart in een doosje zit, daar is de PC-kaart kaal en bloot. Ze worden in de kast gemonteerd, en vormen daarna in principe een vast onderdeel van de machine. Men kan ze wel omwisselen, maar dat is toch wel wat meer werk.

Zonder enige documentatie erbij ontvingen we een lange insteekkaart voor een XT-slot, waarop duidelijk de belangrijkste onderdelen van een MSX aanwezig waren: een Z80 microprocessor, een V9938 videochip, een S3527 MSX-Engine en zelfs een cartridgeslot. Hoewel, die cartridgeslot is een verhaal apart, want hoewel er een vijftigpolige connector op de kaart zit past daar geen cartridge in. Daar hadden we eerst nog een stukje vijftig-aderige lintkabel voor nodig, met een tweetal connectoren. Eentje die op die vijftig pennetjes paste en eentje waar een cartridge op aangesloten kon worden. Logisch ook wel, want om nu de PC open te moeten schroeven om een Konami te spelen, dat is ook niet alles...
Gelukkig werd het gebrek aan gebruiksaanwijzing gecompenseerd door Guus Voet. Guus is een actief PTC-lid, die aan de productie van de eerste serie handgesoldeerde kaarten heeft meegewerkt. Met angst en beven heeft Guus samen met één van onze wat technischer aangelegde redacteuren die lintkabel voor de cartridge gemaakt en vooral aangesloten. Bij een normale MSX is het onmogelijk om een cartridge verkeerd om aan te sluiten, bij deze losse connector was er een statistische kans van vijftig procent. En dan is die cartridge wel naar de eeuwige jachtvelden, als de voedingsspanningen op de verkeerde aansluitingen komen.

De uitvoering

De print zelf ziet er uiterst professioneel uit. Een MSX-Engine – een chip die heel veel functies in zich verenigd – alsmede de dubbelzijdige uitvoering houdt het aantal onderdelen laag. Als we ons even bedenken dat ieder onderdeeltje stuk kan een goede zaak.
Aan de achterzijde van de print – de enige plek waar men bij een PC iets aan kan sluiten zonder schroevendraaier – zijn een tweetal joystick-aansluitingen en een RGB monitoruitgang te vinden. Wat jammer genoeg ontbreekt is de bij dergelijke kaarten toch altijd standaard aanwezige metalen beugel, waardoor de

kaart op zijn plek kan worden vastgeschroefd. De aansluitingen lieten geen ruimte voor dit eigenlijk toch hoogst noodzakelijke onderdeel. Een beetje eng is het wel, zo'n losjes wiebelende kaart. Als de kaart per ongeluk uit de connector in de PC schiet, dan zal dat vrijwel zeker niet ongestraft gebeuren.

Wat ook ontbreekt op de kaart is een printer-aansluiting. Die is dan ook niet nodig, de MSX gebruikt gewoon de PC-printerpoort, die altijd wel aanwezig is. Net zoals de MSX-kaart de seriële PC-poort kan gebruiken.

Het beeld

Eén van de zaken waar we heel benieuwd naar waren was hoe men het monitor-probleem zou hebben opgelost. Immers, de MSX gebruikt als standaard een RGB-monitor, bijvoorbeeld via een SCART-plug. Ook vinden we op de meeste MSX-machines een gewone video-out. PC's kennen hele andere monitoren, volstrekt afwijkend van de RGB-techniek van de MSX.
Men heeft daar een aardige tussenoplossing voor bedacht. Om met de deur in huis te vallen: die MSX-kaart doet het niet op de PC kleurenmonitor. Men heeft – en dat is ergens jammer – een specifieke RGB-monitor nodig. Jammer, omdat de meeste PC-gebruikers zo'n ding dus niet hebben.
Maar het is niet nodig om van de PC een tweekoppig monster te maken. Die ene RGB-monitor kan met een truukje prima het beeld van de PC vertonen. Op de MSX-kaart is namelijk ook voorzien in een ingang, waarmee een extern CGA-signaal – van een video-kaart in de PC – door de MSX kan worden omgezet naar een analoog RGB-signaal. De aansluitkabels – zowel van de MSX-kaart naar SCART als van de CGA-kaart naar de MSX-kaart – worden gelukkig wel bijgeleverd.

Dit biedt opeens ongekende mogelijkheden. Zo kan men met die MSX-kaart opeens een PC aansluiten op een gewone TV, mits die van een SCART-aansluiting voorzien is. Of op een video-recorder, natuurlijk. Het nadeel is echter dat men gebonden is aan de toch wat achterhaalde CGA-standaard. PC-bezitters met een monochrome Hercules-kaart vissen achter het net, evenals diegenen die bijvoorbeeld een fraai VGA-graphics card gebruiken. Het werkt op CGA, en verder niets. Voor de liefhebbers, het kan wel, twee monitoren. Een PC-monitor en een MSX-monitor naast elkaar zetten. En dat heeft toch wel wat, als het PC-beeld keurig blijft staan als de MSX-monitor opkomt.

De software

En daarmee hebben we de hardware gehad. Geen cassette-poort, geen tweede cartridge-slot. Tamelijk rechttoe-rechtaan, maar zeker afdoende.
Maar met alleen die hardware zijn we er nog niet. Om de PC tot MSX om te toveren is er ook software nodig. Op een diskette staan een aantal files, die het uiteindelijke wonder moeten volbrengen. Eén programma – dat je gebruikt terwijl de PC nog niet denkt dat hij een MSX is – slaat een aantal opstellingen op voor gebruik tijdens de emulatie. Er wordt gevraagd of je gebruik wilt maken van de MSX joystickpoorten, van de PC seriële muis of van de PC seriële poort – voor bijvoorbeeld datacommunicatie. Ook vraagt dit programma of de B-drive als RAMdisk moet worden geïnstalleerd. De antwoorden op de vragen worden opgeslagen in een datafile.

Een tweede programma is in staat om communicatie-programma's van Philips zo te patchen dat ze ook werken op de emulatiekaart in samenwerking met de seriële poort van de PC. Handig, heel handig zelfs, want zo kan men ook het PC-modem netjes benutten. Alleen, bij ons werkte het niet, maar dat kan aan het gebrek aan documentatie liggen. Geen letter op papier!
Het derde programma is de eigenlijke emulator. Deze start de MSX-kaart op en zorgt voor de interfacing met de PC-hardware. De drives, de printerpoort, de seriele poort en het toetsenbord van de PC worden door de MSX-emulatiekaart gebruikt, hetgeen inhoudt dat er fiks wat samenwerking tussen de MSX-insteekkaart en de PC moet zijn.

Het moment

Na het insteken en aansluiten van de kaart aan een RGB-monitor dan eindelijk het grote moment: het opstarten! Na het intypen van 'MSX' achter de DOS-prompt verschijnt de volgende boodschap op de PC-monitor:
MSX has been reset.
Simulating MSX I/O...
En even later verschijnt het opstartscherm van onze oude vertrouwde MSX2. De A-drive begint te draaien, de B-drive begint te draaien en even later staat de bekende 'Ok' prompt op het scherm. Een MSX is geboren in het hart van een 16 bits broer, de AT.
We hebben slechts één klein probleempje: onze A-drive is een 5.25 inch drive met een capaciteit van 1.2 Mbyte en onze B-drive is een 3.5 inch drive met een capaciteit van 1.44 Mbyte. Standaard AT-drives dus.
Gewone MSX-programma's worden in het algemeen niet geleverd op één van deze formaten en zelfstartende programmatuur staat altijd op 3.5 inch diskettes. Na de opdracht Call Format hebben we geprobeerd de A-drive te formatteren en dit gebeurde standaard op 720 kByte, het formaat waarmee de gemiddelde MSX-er goed uit de voeten kan. Het formatteren van de 1,44 Mbyte B-drive is tot op heden nog niet gelukt, maar dat zal wel aan ons liggen.

Na de vondst van de te gebruiken A-drive zijn we pas echt aan het testen geslagen. MSXDOS is door ons eerst op de disk in drive A gezet, hetgeen een 5.25 inch 720 Kb MSXDOS disk opleverde. Een ongebruikelijk formaat, maar het werkt. Daarna konden we de meeste programma's ook van de 3.5 inch B-drive opstarten, desnoods door eerst onder DOS van defaultdrive te wisselen.
Het is even zoeken naar de bekende MSX-toetsen <Graph>, <Code>, <Select> en <Stop>, maar als je eenmaal weet dat deze toetsen zijn vervangen door <NumLock>, <ScrollLock>, <+> en <Alt> is er best mee te werken. De cursortoetsen zijn gelukkig gewoon te vinden onder de pijltjestoetsen. De functietoetsen 6 tot en met 10 zijn naar goed MSX-gebruik te vinden met de <Shift> plus <F1> tot en met <F5>. Met de combinatie <Alt> <PrtScr> kom je weer terug in de PC-omgeving, waar dan een viertal keuzemogelijkheden is:

E – exit to PC updating MSX.DTA setup file
Q – quit to PC not updating MSX.DTA setup file
R – Reset the MSX
ESC to continue MSX I/O simulation

Voor de oplettende lezer is het duidelijk dat je dus ook weer uit de MSX kunt komen om verder te werken met 'big brother'.

De opbouw

Om te kijken hoe deze MSX intern is opgebouwd hebben we er eerst ons oude MCM programma MSXMEM op

los gelaten. De resultaten wijzen op een zeer recht toe recht aan Philips MSX-computer:

ROM-BIOS: slot 0-0
MSX1-ROM: slot 0-0
MSX2-ROM: slot 3-0
DISKBASIC: slot 3-3
RAMSLOT: slot 3-2 (als 128k memory-mapper)

Dat de machine ook over 128 kByte Videoram beschikt spreekt vanzelf.

De problemen

Niets is foutloos. Ook deze MSX-insteekkaart vertoont wel wat vreemde foutjes. Zo bleek Red Lights Of Amsterdam, het Aackosoft strippoker programma, muurvast te lopen zodra onze tegenspeelster het woord nam. Waar de gedigitaliseerde spraak zou moeten beginnen liep het beeld in de soep en de machine vast.

Ook qua timing hebben we wel wat problemen gezien, maar vreemd genoeg bleek dat afhankelijk van de PC waar de kaart in gemonteerd was. Op een simpele en niet al te snelle Taiwanese XT-kloon bleek het MCM programma Dsort een onverwacht effect te tonen. Dsort is een directory-sorteerder, waarmee men de volgorde van de bestandsnamen in een directory naar eigen goeddunken kan inrichten. Eén van de mogelijkheden van Dsort is om door de maximaal 112 bestandsnamen heen te scrollen.
Op een MSX loopt dat geheel soepel, op die XT kregen we een soort cyclische vertraging. Iedere paar seconden vertraagde het scrollen, om daarna weer op snelheid te komen. Als klap op de vuurpijl bleek dit euvel op een snelle AT niet voor te komen. Oftewel, timingsproblemen. Althans, dat denken we. De systeemklok van de MSX en de PC lopen natuurlijk op verschillende snelheden, maar moeten wel op de één of andere wijze gesynchroniseerd worden om de onderlinge communicatie mogelijk te maken. En dat lijkt ons de verklaring voor dit vreemde foutje op de XT, wat op de snellere AT niet te dupliceren bleek.

Maar voor de rest: alles draait. Alle programma's – zowel disk als cartridge – die we getest hebben werken probleemloos.
Ease van Philips zelf loopt als een zonnetje, de printer kan daarbij gebruikt worden door hem gewoon op de parallelle printerpoort van de PC aan te sluiten. De onmisbare muis kan zowel aan joystickpoort 1 van de MSX als aan de seriele poort van de PC worden gehangen. Je moet dan wel even in SETMSX vertellen welke poort je wil gebruiken en je moet er rekening mee houden dat aan de seriele poort van de PC alleen een PC-muis kan, maar dat is eigenlijk wel logisch. We vonden wel dat de muis die direct aan de MSX hangt iets soepeler werkt, de PC-versie werkt een beetje schokkerig. Alweer timing-probleempjes, vermoeden we.

Ook cartridges draaien naar behoren. Gewoon in het cartridgeslot doen – liefst niet verkeerd-om – en de PC (en dus de MSX) aanzetten. Nemesis, King's Valley, noem het maar op. Dingen die niet werkten hebben we niet gezien.
Navraag bij de PTC hoe het allemaal zo goed kon werken leverde een kort, maar duidelijk antwoord op: het is een MSX2 met een gewone MSX2 ROM. Het lijkt ons dan wel logisch dat een en ander naar behoren werkt.

De kritiek

Ergens is het jammer dat de MSX-kaart niet de bestaande PC-monitor kan benutten. Dat maakt de mogelijkheid om met deze kaart PC-gebruikers aan de MSX te helpen wel wat kleiner. Alleen, hoe men technisch had moeten realiseren om de PC-monitor – en welke dan wel, Hercules, CGA, EGA, VGA – op de MSX videochip aan te sluiten is ons ook niet zonneklaar. Dit compromis is waarschijnlijk toch de beste oplossing.

Daarnaast hadden we stilletjes gehoopt de PC-harddisk te kunnen gebruiken. Maar dat ging jammer genoeg niet door. Spijtig, spijtig, maar ook hier geldt dat de technische problemen waarschijnlijk wel wat erg groot waren.

Op sommige PC's geeft de kaart problemen bij het spelen van spellen waarbij snel gescroll't wordt. Op de AT waarop we de emulatiekaart hebben getest ging het allemaal vlekkeloos, scrollende spelen zagen er gewoon goed uit. Op sommige XT's zou zich dit probleem voor kunnen doen. Waarschijnlijk wordt dit veroorzaakt door de al vaker genoemde timing-problemen, die ontstaan omdat twee geheel verschillende kloksnelheden – die van de MSX en die van de PC – toch op elkaar moeten aansluiten.

Bij de PTC hadden ze hier niets van gemerkt, navragen leverde op dat ze daar ook een snelle AT als gastheer gebruiken.

De conclusie

Prachtig. Een wondertje der techniek. Ondanks de kleine probleempjes die we ondervonden een uitkomst voor iedereen die zowel MSX als PC gebruikt. En bovendien een leuke oplossing voor PC-gebruikers die wel eens wat meer willen kunnen dat een tekstverwerker of een spreadsheet.

Alleen, wel te laat. Deze kaart had twee, drie jaar geleden op de markt moeten komen. Philips heeft hier een steek laten vallen, toen men de ontwikkeling stillegde. Eeuwig jammer dat we zo lang hebben moeten wachten op deze MSX-insteekkaart.
Dat doet echter niets af aan ons eerlijk enthousiasme. Een inbouw MSX voor een alleszins redelijke prijs, met de nodige kabels en – zo is ons beloofd – een handleiding.

Maar een paar spijkers op laag water weten we altijd wel te vinden. Zo denken we dat het een misser is dat de kaart straks zonder de kabel om het slot naar buiten te voeren zal worden geleverd. De onderdelen zijn niet zo duur en ook voor vijfentwintig gulden meer was de kaart nog heel redelijk geprijsd geweest. Nu moet men zelf op zoek naar kabel en connectoren, om die vervolgens aan elkaar te zetten en in de goede orientatie aan te sluiten.
Ook jammer is de nieuwe politiek van de Philips Thuiscomputer Club, om allerlei zaken – zoals deze kaart en de elders besproken NMS 1436 printer – alleen aan leden te verkopen. Eén van onze redacteuren heeft dat een tijdje terug al mogen ondervinden, toen er een aantal MSX-computers via de PTC aan de leden werden aangeboden. Nadat de brave borst hals over kop lid geworden was bleken de computers net uitverkocht. Maar goed, nu kan hij dus de insteekkaart bestellen.

MSX-insteekkaart voor PC
Prijs: *f* 599,-
Het verplichte PTC-lidmaatschap kost *f* 46,- per jaar.

Bestellen en nadere informatie:
Tel.: 040-758912
Bellen op werkdagen tussen 10.00 en 12.00 of 13.00 tot 16.00 uur.



COMPUTERSPELLEN

Darwin 4078

'Darwin 4078' is een door Hudson Soft in licentie vervaardigd spel. De rechten berusten bij Data East Corporation en dat betekent dat dit spel oorspronkelijk een origineel amusementshal spel moet zijn. Tot mijn spijt moet ik zeggen dat ik dit spel nog nooit in een Nederlandse hal gezien heb. Dientengevolge kan ik ook niet bepalen hoe goed de conversie naar MSX2 is. Afgaand op deze conversie ben ik echter blij dat ik het nooit gezien heb; de guldens die dit spel gekost zou hebben!

'Darwin 4078' is verbazingwekkend goed. In alle opzichten is deze verticaal scrollende 'shoot'em up' af. Grafisch is alles perfect, met slechts nauwelijks waarneembare flikker wanneer meerdere objecten tegelijkertijd razendsnel over het scherm bewegen. De achtergrondgraphics zijn gestoken scherp en uiterst gedetailleerd. Ook qua geluid valt er weinig aan te merken, het is alleen jammer dat het FM-PAC niet ondersteund wordt.

Het is overduidelijk dat 'Darwin 4078' een Data East Corp spel is. Alle nadruk ligt op messcherpe arcade-actie, het kenmerk van alle Data East amusementshal spellen. Eerlijkheidshalve moet ik toegeven dat het geheel me toch ietwat teleurstelde, gewend als ik ondertussen ben aan de grappen en grollen die Japanse programmeurs in te schijnen moeten bouwen in hun spellen. En die heb ik tot nog toe niet in dit klassespel mogen ontdekken.
Er gaan echter geruchten dat ik wat betreft 'Darwin 4078' toch te somber ben, want er schijnen wel degelijk geheime velden en dat soort grappen in voor te komen. Ik hou me aanbevolen, want ik ben ze nog niet tegengekomen. De E.H.B.O.'er heeft toegezegd brieven over 'Darwin 4078' onmiddellijk en met voorrang te plaatsen, ook hij schijnt menig uurtje met dit spel doorgebracht te hebben!

Verder kun je over 'Darwin 4078' heel erg kort zijn. Een uitstekend spel, dat iedere fanatieke vuurknopfreak tot waanzin zal drijven. Schieten, schieten en nog meer schieten is het enige dat je in dit spel overeind houdt. Waarbij zowel de grafische vormgeving als afwerking alleen maar bijdragen aan het actieplezier.

Voor duimen met vuurknop-eelt aanbevolen!

SPELLEN GETEST VOOR DE LIEFHEBBERS





Fabrikant: Hudson Soft
Importeur: HomeSoft

Computer: MSX2
Medium: MegaROM
Aantal spelers: 1
Bediening: joystick & toetsenbord
FM-PAC Muziek: nee; S-RAM: nee
Prijs: *f* 119,-

Mon Mon Monster

De Japanse import 'The Mon Mon Monster' is weer eens voorzien van zo'n titel welke in Europa totaal niets betekent, maar in Japan een begrip schijnt te zijn. De verpakking sluit echter aanzienlijk beter aan op de Europese belevingswereld. Het 'Monster van Frankenstein' grijnst ons vervaarlijk toe op de hoes. En daarmee hebben we dan meteen de hoofdpersoon in dit vrij recente, van eind 1989 daterend MSX2/2+ spel te pakken: het 'Monster van Frankenstein'.

Qua spel hoort 'The Mon Mon Monster' thuis in het rijtje 'Super Wonder Boy' (nog steeds niet voor MSX), 'Psycho World', 'Vampire Killer', 'Fantasm Soldier – Valis', 'Contra' en 'Sa-Zi-Ri' om maar eens een paar titels te noemen. Derhalve een horizontaal scrollend platformspel waarin de hoofdpersoon moet zien te overleven, onderwijl voorwerpen oppikkend en deze gebruikend om verder in het spel te kunnen doordringen. In 'The Mon Mon Monster' geldt dit alles ook, alhoewel als 'Monster van Frankenstein' je natuurlijk wel wat meer in je mars hebt. Met behulp van vuurknop A spring je, terwijl vuurknop B je stevige knuisten het zwaarste rotsblok aan flarden laat slaan. Sterker nog, vervaarlijke lichtflitsen schieten uit je vuisten wanneer je deze gebruikt, zodat je ook monsters op afstand kunt vernietigen.
De boksbewegingen van de heldsprite zijn overigens uitermate fraai uitgevoerd, weer eens een bewijs temeer hoezeer Japanse MSX-programmeurs hun vak beheersen.

Wat het uiteindelijke doel van dit spel is, is me enigzins onduidelijk. Dat je kunt springen en slaan was meteen duidelijk. Ook dat je slechts drie levens ter beschikking hebt of dat je één van die kostbare levens verliest nadat je drie keer door iets getroffen bent, was niet al te moeilijk. Wat echter sommige voorwerpen zoals schroeven en laarzen kunnen betekenen moet me nog eens uitgelegd worden. Andere voorwerpen zijn duidelijk: extra kracht, extra levens of de mogelijkheid om grote rotsen naar je tegenstanders te gooien leer je al snel te onderkennen.
Zoals maar al te vaak tegenwoordig ging dit recensie-exemplaar niet van een Engelse of Nederlandse handleiding vergezeld. Toch is dit niet echt een ramp, het spel is goed te spelen alhoewel er, zoals vermeld, wel enig gepuzzel aan te pas zal komen om de betekenis van alle voorwerpen te kunnen vinden.
Het blijft echter puzzelen. Pas toen ik door een deur heengegaan was – en ik even dacht niveau twee te betreden – bleek ik me in de tweede laag van het eerste niveau te bevinden. Je kunt overigens zien waar je in het spel bent door de STOP toets in te drukken. In het beeld verschijnt een kaart waarop je kunt zien hoeveel velden je al gepasseerd bent, waar je nu bent en waar in het niveau het zogenaamde 'BOSS' monster zich bevindt.

Wat me ook pas relatief laat duidelijk werd is het feit dat je niet alleen de op je pad liggende vierkante rotsblokken kunt – ja, zelfs moet – verbrijzelen.
Op gegeven moment liep ik in een kuil tegen de muur op onderwijl zenuwachtig de vuurknop beroerend. Nu was ik deze plek al vaker gepasseerd – het was niet mijn eerste poging – maar tot mijn niet geringe verbazing baande ik me een weg door de rotsen heen die tot dan volledig ondoordringbaar leken. Halverwege het scherm vond ik zelfs een flesje levenswater.
Deze ontdekking bleek zeer belangrijk. Er zijn namelijk bepaalde locaties in het spel die niet bereikbaar lijken maar waar je je, na enig zoeken via een omweg – en pure, botte kracht – naartoe moet rammen.

Al snel bleek dat 'The Mon Mon Monster' meer is dan het zoveelste loop&doe spel. Het spel bevat zoveel mogelijkheden om de velden te doorlopen dat zonder de mogelijkheid een kaart op te roepen je al snel totaal zou verdwalen. En dan moet ik nog niet eens denken aan alle ondoordringbaar lijkende gedeelten; wie weet wat daar nog verstopt zit?

Met 'The Mon Mon Monster' blijkt weer eens volop dat de Japanse spellencultuur totaal anders is dan de Europese. Waar hier de botte schietspellen of arcade adventures hoogtij vieren zijn de Japanse spellen complete kunstwerken. Een op het eerste gezicht simpel schietspel als Nemesis blijkt veel meer te bevatten, hetzelfde geldt voor 'The Mon Mon Monster'. Niet voor niets zien we dan ook dat oorspronkelijk in Japan ontwikkelde spellen met name de Amerikaanse markt – en in mindere mate de Europese – veroveren.

lees verder op pagina 26

MSX: LIST-extra

Wie in Basic programmeert zal regelmatig het LIST-commando gebruiken. En daarbij vrijwel even vaak de gewenste regels in hoog tempo weer van het scherm af zien rollen. Steven van Loef, één van onze vaste inzenders, bedacht de oplossing voor dit euvel. Geen taalvervuiling meer onder het programmeren, afgelopen met het gevloek.
Maar eerst zullen we de precieze werking van het LIST-commando in MSX-Basic eens uitleggen.

Het LIST-commando is mogelijk wel het meest gebruikte Basic-commando. Wie al wat meer heeft geprogrammeerd kent de syntax op zijn duimpje:

LIST [regelnummer] [- regelnummer]

Gewoon LIST betekent dat het hele programma, van begin tot eind, over het scherm heen komt denderen. Met STOP kan men dat even stilzetten, om desgewenst met CONTROL-STOP uit het listen terug te vallen in de gewone editor.

Regelnummers

Na het LIST-commando kan men ook een regelnummer opgeven. In dat geval zal LIST slechts die ene regel laten zien. Een punt mag ook, daarmee geven we dan aan dat we de huidige regel willen zien.

Voor Basic is er namelijk altijd een huidige regel. Als we een programma bijvoorbeeld afbreken, dan gebeurt dat altijd in een bepaalde regel. Dat regelnummer wordt bewaard en kunnen we met die punt aangeven. Makkelijk, bij het ontluizen van een programma.

Om vanaf een bepaalde regel te gaan listen, daar hebben we het min-teken voor. LIST .- betekent zoveel als 'list het programma te beginnen met de huidige regel tot en met het einde'.
Het eventuele tweede regelnummer geeft aan tot waar we willen listen. Met een commando zoals:

LIST 100-200

krijgen we de regels vanaf nummer 100 tot en met 200 op het scherm. Mocht er geen regel 100 bestaan, geen nood. LIST begint dan met de eerste regel met een regelnummer hoger dan 100.

Primitief

Enerzijds is het LIST-commando goed op zijn taak toegesneden, anderzijds heeft het toch iets primitiefs. Want stel

EXTRA BASIC-COMMANDO VOOR MSX

Excuses voor de wat vreemde plaats van deze bon

I/O'tjes

Gratis voor abonnees van MSX Computer Magazine.

Zoekt u iets, of heeft u iets aan te bieden op computergebied? Plaats dan een I/O'tje.

IK ZOEK **IK BIED AAN**

Ik ben abonnee sinds:
Ik ben geen abonnee en heb een geldig betaalmiddel van 5 gulden bijgesloten
Invullen in duidelijke blokletters, verzenden in gesloten envelop naar:
Wegener Tijl Tijdschriften Groep b.v.
Postbus 9943, 1006 AP Amsterdam

Afz: . . . .
Adres: . . . .
Postcode: . . . .
Plaats: . . . .
Tel: . . . .

je eens voor dat een gewone tekstverwerker op een dergelijke manier bediend zou moeten worden, om een bepaald stuk tekst op het scherm te brengen. Even een regeltje omhoog of omlaag, dat zou ideaal zijn.
Want wat is de praktijk van alledag, men weet nooit precies welke regels men nu precies wil zien.

Oftewel, LIST wordt misschien wel met een beginregel aangeroepen, maar die eindregel vergeten de meeste programmeurs maar. Op tijd op STOP drukken, terwijl de listing over het scherm rolt, dat is de dagelijks gebruikte truuk. En dat gaat dan ook maar al te vaak fout, waarna de beschaafde programmeur volstaat met een welgemeend 'shit'.

Extra's

Dit is nu precies waar Steven's nieuwe utility op inspeelt. Het is een kleine uitbreiding op het LIST-commando, waarmee we voortaan net even wat meer gemak hebben.
Na:

* LIST regelnummer – regelnummer

verschijnt het gevraagde stuk programma keurig scherm voor scherm. na ieder scherm pauzeert de MSX, tot men op de spatiebalk drukt. Met de cursor omhoog en omlaag kan men zelfs één regeltje op het scherm verspringen. De Escape tenslotte beeindigt deze luxe vorm van LIST, waarna men weer gewoon in de Basic-omgeving belandt.

Deze Basic-uitbreiding is natuurlijk in machinetaal geschreven. Het programma installeert deze boven in het vrije geheugen en zet de pointer voor het beschikbare Basic-geheugen meteen goed. Oftewel, de LIST-uitbreiding moet voordat men aan het werk gaat geladen worden, anders wordt het programma in het geheugen vernield. Erg veel ruimte neemt één en ander niet, nog geen 180 bytes.

Al met al een leuke extra voor MSX Basic programmeurs, dit programma van Steven van Loef.

```
1Ø REM LIST                                   Ø
2Ø REM                                        Ø
3Ø REM LIST-commando uitbreiding              Ø
4Ø REM                                        Ø
5Ø REM Copyright 1988 by:                     Ø
6Ø REM                                        Ø
7Ø REM S.V.L. Software Vleuten                Ø
8Ø REM                                        Ø
9Ø CLEAR 2ØØ,PEEK(&HFC4A)+256*PEEK(&H
FC4B)-&HB3                                    45
1ØØ W=PEEK(&HF3BØ)                            16Ø
11Ø SCREEN Ø: WIDTH 39                        7
12Ø CLS: LOCATE 7,9: PRINT "LIST-comm
ando uitbreiding"                             135
13Ø PRINT: PRINT TAB(12);"By: S. van
Loef"                                         77
14Ø PRINT: PRINT TAB(8);"Machinecode
installeren"                                  224
15Ø ST=PEEK(&HFC4A)+256*PEEK(&HFC4B)          194
16Ø FOR I=Ø TO &HB2                           158
17Ø   READ A$                                 166
18Ø   CS=CS+VAL("&H"+A$)                      213
19Ø   IF A$="*" THEN READ A$: CS=CS+V
AL("&H"+A$): P=ST+VAL("&H"+A$): POKE
ST+I,P-INT(P/256)*256: I=I+1: POKE ST
+I,INT(P/256) ELSE POKE ST+I,VAL("&H"
+A$)                                          49
2ØØ NEXT I                                    2ØØ
21Ø IF CS<>226Ø4 THEN CLS: PRINT "U h
eeft een fout gemaakt in de data-    r
egels!": PRINT: PRINT "Eerst verbeter
en!": STOP                                    142
22Ø POKE &HFF89,&HC3                          129
23Ø POKE &HFF8A,ST-INT(ST/256)*256            51
24Ø POKE &HFF8B,INT(ST/256)                   3Ø
25Ø CLS: WIDTH W: PRINT "ML geinstall
eerd op ";HEX$(ST)                            16Ø
26Ø PRINT: PRINT "Gebruik LIST als vo
lgt:"                                         167
27Ø PRINT: PRINT "* LIST regelnummer
- regelnummer"                                18Ø
28Ø PRINT "* LIST -"                          81
29Ø PRINT: PRINT "Tijdens het listen
zijn de volgende    commando's beschi
kbaar:"                                       182
3ØØ PRINT: PRINT "[ESC]          - Ter
ug naar command-mode"                         35
31Ø PRINT "cursor omhoog - Volgende r
egel"                                         98
32Ø PRINT "cursor omlaag - Vorige reg
el"                                           146
33Ø PRINT "spatie         - Volgende b
ladzijde"                                     22Ø
34Ø END                                       18Ø
35Ø ' ML-listing *******************          Ø
36Ø DATA 3A,16,F4,FE,Ø1,C8,3A,DE,F3,3
2,*,BØ,CD,CC,ØØ,3E,ØC,DF,C1,CD,79,42,
ED,43,*,B1,C5,21,FF,FF,22,1C                  167
37Ø DATA F4,E1,D1,4E,23,46,23,78,B1,2
8,7B,C5,4E,23,46,23,C5,E3,EB,E7,C1,38
,6F,E3,E5,C5,EB,22,B5,F6,CD,12                91
38Ø DATA 34,E1,7E,FE,Ø9,28,Ø3,3E,2Ø,D
F,CD,84,52,21,5E,F5,CD,7B,52,CD,28,73
,3A,DC,F3,FE,14,38,BE,CD,9F,ØØ                65
39Ø DATA FE,1B,28,4Ø,FE,1E,28,12,FE,1
F,28,19,FE,2Ø,2Ø,ED,3E,ØC,DF,E1,E5,22
,*,B1,18,A1,E1,2A,*,B1,7E,23                  198
4ØØ DATA 66,6F,E5,18,EB,ED,4B,*,B1,6Ø
,69,ED,5B,76,F6,1B,2B,E7,28,C9,78,BE,
2Ø,F8,2B,E7,28,C1,79,BE,2Ø,F1                 29
41Ø DATA D1,E5,18,CC,E1,E1,3A,*,BØ,B7
,C4,CF,ØØ,C3,1F,41,ØØ,ØØ,ØØ                   1Ø9
```



Klaarblijkelijk is men daar nu ook rijp voor dit soort ingewikkelde spellen. Misschien valt hier voor een gedeelte het grote succes van de Nintendo en Sega spelcomputers in Amerika uit te verklaren. In Europa overheersen op dit moment nog de Engelse software huizen waar de nadruk nog steeds op bot schietwerk ligt, alhoewel hier recentelijk enige verandering in te bespeuren is.

Qua graphics en muziek (FM-PAC uiteindelijk) valt er weinig over 'The Mon Mon Monster' te zeggen: klasse gewoon! De groene aapjes met hun uitschuifnekken zien er perfect uit en de eerste keer dat een standbeeld aan de wandel ging nadat ik hem aan gruzelementen probeerde te slaan blijft me nog steeds bij.

Aanrader!

Fabrikant: Hot B Co.
Importeur: HomeSoft

Computer: MSX2/2+
Medium: MegaROM
VRAM: 128k
Aantal spelers: 1
Bediening: joystick & toetsenbord
FM-PAC Muziek: ja; S-RAM: nee
Prijs: *f* 119,-

Famicle Parodic

De uitsluitend voor MSX2 geschikte Mega ROM 'Famicle Parodic' is al weer een paar maanden verkrijgbaar. Toegegeven, we hebben deze ROM ietwat over het hoofd gezien. In de FM-PAC euforie zijn we deze ROM van het niet onaanzienlijke Japanse huis Bit2 gewoonweg vergeten. En dat verdient deze Mega ROM zeker niet.

In wezen is 'Famicle Parodic' een eenvoudig schietspel, ware het niet met een overduidelijke Japanse inslag. Waar een Engels software huis een recht voor z'n raap schietspel geproduceerd zou hebben heeft Bit2 zoveel toegevoegd dat we niet meer van een schietspel sec kunnen spreken.
Niet alleen bepaal je met welke kapitein je wilt strijden, ook het soort toestel bepaal je zelf. Zo kun je kiezen uit Papicle met z'n Voller Bulle, Mamicle met de Mine Mime, Mirocle met de P.P. Kerol, Silucle met Ducks Berry en Nyancle met Ciel Chatte. Opvallend is overigens dat het merendeel van deze helden dames schijnen te zijn.
Hoewel het spel grotendeels schieten is kent het ook een aantal andere aspecten. Zo kun je bij tijd en wijle in de zogenaamde 'shop' – de winkel – wapens kopen c.q. verkopen. In dat opzicht doet 'Famicle Parodic' denken aan het beroemde Engelse spel voor de Atari ST en Amiga, Xenon 2. Het zou me zelfs niet eens verbazen wanneer de Bitmap Brothers, de programmeurs van Xenon 2, hun ideeën aan 'Famicle Parodic' ontleend hadden.
Daarnaast vlieg je over de meest fantastische landschappen, variërend van het golflandschap in veld één tot het waanzinnige puzzlestukjeslandschap verder-op in het spel. Ook 'Famicle Parodic' is weer eens een voorbeeld van hoe de Japanners tegen spellen aankijken. Een spel mag dan simpel lijken, onderhuids is er zoveel mogelijk en nodig om het spel uit te spelen dat je moeilijk meer van een simpel spel kunt spreken. De Engelse software huizen kunnen hier een voorbeeld aan nemen.
Qua graphics is 'Famicle Parodic' van de standaard, zeer hoge Japanse kwaliteit. Kleurrijk, vloeiende bewegingen, goed scrollend en boordevol actie waarbij het jammer is dat het FM-PAC niet wat betreft geluid ondersteund wordt. Wel is het mogelijk het S-RAM van de FM-PAC te gebruiken.

Voor de ware actie/puzzle fanaat is 'Famicle Parodic' een verplichte aanschaf.

Fabrikant: Bit2
Importeur: HomeSoft

Computer: MSX2
Medium: MegaROM
VRAM 128k
Aantal spelers: 1
Bediening: joystick & toetsenbord
FM-PAC Muziek: nee; S-RAM: ja
Prijs: *f* 129,-

Yajikita

Het uit 1987 daterende spel 'Yajikita' is een schoolvoorbeeld van hoe goede bedoelingen de mist in kunnen gaan.
Bij het openmaken van de verpakking wacht de eerste verrassing. In de doos zitten een dobbelsteen, vier speelstukken en een speelbord – oftewel de achterkant van de handleiding. Het Japans vliegt je om de oren. Wat is hiervan nu weer de bedoeling?

Op het eerste gezicht lijkt 'Yajikita' een spel van het Hydlide of Ash-Guine type. In bovenaanzicht wandel je door een dorp en je komt verschillende dorpelingen tegen, met wie je een praatje aangaat door tegen ze op te lopen en vuurknop B in te drukken. Jammer genoeg is het uitvoerige gesprek uitsluitend in het Japans te genieten, wat je dus medegedeeld wordt is totaal onduidelijk. Ook in de huizen en winkels wordt één en ander van je verwacht, maar het gebrek aan een goede, vertaalde handleiding plus het overvloedige Japans op het scherm staan er garant voor dat je totaal geen kant uit kunt. Wat reden is geweest dat HomeSoft besloten heeft dit, toch al wat oudere spel te importeren is me een



raadsel. Een spel uit 1987 mag van mij als het de moeite waard is – ook al ondersteunt het de FM-PAC op generlei wijze. Persoonlijk ben ik niet zo gespitst op het nieuwste van het nieuwste. Liever zie ik een goed spel dan het nieuwste flutspel, waarbij het flut zijn overschaduwd wordt doordat het **nieuw** is. De inhoud telt dan opeens klaarblijkelijk niet meer. Neen, de uiterst snobistische opstelling van 'het is alleen maar goed als het de nieuwste Japanse import betreft' heb ik per definitie niet!
Maar goed, misschien is 'Yajikita' wel de moeite waard. Op zich vind ik de combinatie bordspel/computerspel best uitdagend. Alleen, wist ik maar wat de **bedoeling** is!
De hoeveelheid schermtekst is dermate groot dat geen normaal mens – althans de hier gebruikelijke, geen Japans beheersende variëteit – hier uit kan komen. Ook het speelbord schijnt een belangrijke functie te hebben, **maar welke?** En wat moet ik met die dobbelsteen en die bordstukken? Raadsels volop en geen oplossingen.

Grafisch ziet 'Yajikita' er verzorgd uit, beter dan we van HAL gewend zijn. Qua geluid valt er weinig op dit spel aan te merken, behalve dan dat dit een pre FM-PAC product is.
Tja, wat moet je nu met zoiets? Ik apprecieer het dat iemand uit Japan importeert, zeker wanneer men probeert dit op regelmatige en professionele basis te doen en niet zoals op dit moment zoveel gebeurt te hooi en te gras. Met dat laatste schiet niemand iets op.
HomeSoft heeft bewezen een goede neus voor titels te hebben: Contra, Cockpit, Ikari Warriors, R-Type, het FM-PAC, Psycho World en ga zo maar even door. Met 'Yajikita' vliegen ze even grandioos uit de bocht.
Voor de masochist! Afblijven!

Fabrikant: HAL Laboratories
Importeur: HomeSoft

Computer: MSX2
Medium: MegaROM
RAM: 64k
Aantal spelers: 1
Bediening: joystick & toetsenbord
FM-PAC Muziek: nee; S-RAM: nee
Prijs: *f* 99,50

Bandit

Naast alle Japanse importen dit keer ook een in Nederland ontwikkeld spel. Bandit, van de ACME Software Company, uitgebracht door New Dimension Software in Harderwijk.
Volgens de tekst op het doosje hebben we te maken met een van de betere gokkasten ooit gemaakt. Een fruitautomaat volgens het Twin-Spin-principe, compleet met een bonus-spel.

Ongetwijfeld waar. De laatste keer dat ik een gokkast heb bespeeld ligt al jaren achter me. En ook toen was dat zo min mogelijk gokken, maar wachten tot iemand er de nodige guldens aan gespendeerd had om daarna – met een wat optimistische inschatting van de wetten der kansberekening – zelf op jacht naar de jackpot te gaan. Die ik overigens nooit gewonnen heb...
Bandit heeft op zijn minst een groot voordeel boven de in de horeca gebruikelijke eenarmige bandieten. Er zit namelijk geen inworpgleuf in. Het kost de speler, na de aanschaf van het spel zelf natuurlijk, verder geen geld. Voor de rest, een redelijk standaard gokkast. Fraaie sprites, die echter niet helemaal natuurgetrouw in de vensters aan de speler voorbijtrekken. De rollen vertragen namelijk niet, wat ze bij de echte gokkasten nu net wel doen.

Dat maakt de spanning wel minder, want bij de gewone gokkast draaiden die rotdingen altijd nu net een tikje verder dan je hoopte. Bij Bandit komen de rollen vanuit vliegende vaart plotsklaps tot stilstand.
Ook het geluid is wat magertjes, maar kan volgens de gebruiksaanwijzing uitgeschakeld worden.

Al met al een spel dat mij niet echt kan boeien, maar dat zegt nog niets over de kwaliteit van het programma. De grafische verzorging is prima, het speelt precies zoals men zou mogen verwachten. Alleen, echt spannend is het niet. Gokken verliest een flink deel van zijn aantrekkingskracht zodra men niets kan winnen, behalve punten die zich niet bij de verkoper van Bandit tegen klinkende munt laten inwisselen.

Wie een ziekelijk gokker is zou bij Bandit mogelijk soelaas kunnen vinden. De sigaret losjes tussen de lippen, het pilsje in de ene hand en spelen maar. Aangezien bier bij de slijter goedkoper is dan in de kroeg wordt het leven opeens een stuk goedkoper.

Fabrikant: ACME Software Company
Leverancier: New Dimension Software

Computer: MSX2
Medium: 3.5 inch disk
RAM: 64k
Aantal spelers: 1
Bediening: toetsenbord
FM-PAC Muziek: nee; S-RAM: nee
Prijs: *f* 24,95



De NMS 1436 Multicompatible printer

Wat? Een recensie van een printer die al een tijd van de markt is? Jazeker, want van dit prima MSX-printertje is onlangs weer een partij boven water gekomen. Een Philips-printer – de NMS 1436 – die zowel MSX als PC is. Nu zijn PC-printers niet zo bijzonder, maar een heuse MSX-printer , voor die prijs, zijn zeldzaam, tegenwoordig.



Weliswaar zijn er heel aardige Japanse import-apparaten, maar die zijn allemaal 24-naalds, hetgeen wel een prima kwaliteit waarborgt maar de prijs aan de hoge kant houdt. Vandaar dat we deze NMS 1436 als een goedkope MSX-printer eens flink in het zonnetje zetten. Ook al is het de laatste restpartij, het is qua prijs en kwaliteit een heel aardig aanbod.

Printers zijn er in alle soorten en maten. Allemaal hebben ze hun eigen besturingscodes, hun eigen tekenset en soms ook nog een eigen aansluiting. Een nieuwe printer zal meestal niet zonder meer direct op een bepaalde computer werken, tenminste, als we alle mogelijkheden willen kunnen gebruiken. Vooral bij MSX-computers geldt dit, want de normale IBM-compatible printer heeft een nu net afwijkende tekenset. Gewoon tekst afdrukken gaat prima, maar verwacht niet dat de bijzondere tekens er goed uitkomen.
Toen er nog genoeg MSX-printers in de winkel stonden, was dat geen probleem. Soms werd dit probleem opgelost door zogenoemde PIC's – Printer Interface Cartridges – toe te passen. Dit zijn insteekmodules met besturings-electronica die ervoor zorgen dat de printer op een bepaald type computer draait. Behalve de printer moest er dus ook nog de passende interface worden aangeschaft.
Philips heeft in de NMS 1436 Multicompatible voor een betere oplossing gezorgd. Deze printer is standaard voorzien van flink wat extra intelligentie, zodat de printer zich aan verschillende computers kan aanpassen. Hij heeft vijf verschillende tekensets aan boord en kan in vier besturings-modes worden aangesproken.
De printer is hierdoor zowel op een MSX als op een PC aan te sluiten. Een ideale machine dus voor diegenen die beide computers in huis hebben.

VEELZIJDIG EN GOEDKOOP

Verschillende modes

In de MSX-mode is de printer geheel compatibel met de voorgangers, de VW-0030 en de NMS 1431. Alle besturingscodes zijn gelijk gebleven. Er zijn echter twee codes bijgekomen. Deze dienen om de MSX-tekenset of Prestel videotex-tekenset te kiezen. Het printvoorbeeld laat zien wat er zoal mogelijk is.
In de andere modes is het aantal mogelijkheden belangrijk uitgebreider. De meeste commando's hebben betrekking op het kiezen van de tekenset of dienen voor een fijnere besturing van de printkop. De Vertical Tabs bijvoorbeeld moeten we in de MSX-mode node missen.
In de Epson- en Prestel-mode kunnen er bovendien 128 download-characters in het printer-RAM worden opgeslagen. Dit zijn zelf te definiëren tekens, die als extra tekenset kunnen dienen.

De besturingscodes voor de IBM-, Epson- en Prestel-mode wijken onderling veel minder af dan van de MSX-codes. In het handboek zijn ze bij elkaar ondergebracht. Er zijn twee gedeelten: één voor MSX en één voor de andere modes.

Het is overigens zaak goed op de mode te letten waarin men werkt. De commando's kunnen per mode een totaal verschillend resultaat opleveren. In sommige modes hebben bepaalde commando's geen effect.

```
** MSX COMPUTER MAGAZINE PRINTERTEST **
   PHILIPS NMS1436 Multicompatible

De MSX tekenset:

  ☺ ☻ ♥ ♦ ♣ ♠ • ◘ ○ ◙ ♂ ♀ ♪ ♫ ☼ ┼ ┴ ┬ ┤ ├ ┼ │ ─ ┌ ┐ └ ┘ ╳ ╱ ╲ ╋   ! " # $ % & '
( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m n o p q r s t u v w
x y z { ¦ } ~ ▵ Ç ü é â ä à å ç ê ë è ï î ì Ä Å É æ Æ ô ö ò û ù ÿ Ö Ü ¢ £ ¥ ₧ ƒ
á í ó ú ñ Ñ ª º ¿ ⌐ ¬ ½ ¼ ¡ « » Ã ã Ĩ ĩ Õ õ Ũ ũ Ĳ ĳ ¾ ∽ ◇ ‰ ¶ § ▂ ▚ ▆ ▔ ◾ ▇ ▎ ▞
▊ ▍ ▋ ▌ ▎ ▏ ⁄⁄ ⁄⁄ ▼ ▲ ▶ ◀ ⧗ ⧓ ▘ ▗ ▝ ▖ ▒ Δ ‡ ω █ ▄ ▌ ▐ ▀ α β Γ π Σ σ µ τ Φ θ Ω δ ∞ ∅ ∈ ∩
≡ ± ≥ ≤ ⌠ ⌡ ÷ ≈ ° ∙ · √ ⁿ ² ■

De Prestel (Videotex) tekenset:

  ! " £ $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G
H I J K L M N O P Q R S T U V W X Y Z ← ½ → ↑ # — a b c d e f g h i j k l m n o
p q r s t u v w x y z ¼ ‖ ¾ ÷ ■
```

Een overzicht van de mogelijkheden:

Pica: 10 tekens per inch of Elite: 12 tekens per inch
Condensed is klein, maar fijn: 17 tekens per inch
In proportioneel schrift heeft elk teken zijn eigen breedte
Letter-quality geeft een fraai lettertje

Ook de kantlijn kan worden ingesteld:

Bold is vet printen maar **Double strike geeft ook meer zwarting**

Dubbele breedte: 40 tekens op een regel

Cursief en <u>onderstreept</u> *<u>trekken de aandacht</u>*

voor formules is er superscript: $a^2+b^2=c^2$ en subscript: H_2SO_4

Enkele combinaties:

Letter-quality cursief   ***Cursief en bold***

Double strike en bold is dubbel vet

Letter-quality en breed

Breed en cursief Elite

De regelafstand is ook instelbaar. Bijna elke gewenste afstand is mogelijk.

de regels komen steeds dichter op elkaar

de regels komen steeds dichter op elkaar

de regels komen steeds dichter op elkaar

de regels komen steeds dichter op elkaar

Grafische tekens zijn zelf te definieren en met verschillende dichtheid af te drukken. Bijvoorbeeld:

Het afgebeelde printvoorbeeld is tot 90% verkleind

Papiersoorten

De NMS 1436 kan met twee soorten papier overweg: losse vellen en kettingpapier. Het invoeren van losse velletjes verloopt heel soepel. Een vel wordt tussen de geleiders op het papierplateau gelegd en wordt dan door het naar voren halen van de invoer-handle automatisch in de goede uitgangspositie geplaatst. Er kan eventueel ook met een sheetfeeder – een magazijn met losse vellen – worden gewerkt.

Voor het gebruik van kettingpapier is een zogenoemde tractor-unit nodig, die standaard wordt meegeleverd. Deze tractor leidt het papier via pennetjes die in de perforatie langs het papier passen door de printer. Het papierplateau kan plat worden gelegd en fungeert dan als papierscheider tussen de in- en uitgaande papierstroken.

Het is jammer dat de printer geen parkeerstand heeft, om bij het gebruik van

kettingpapier toch even een los velletje is af te drukken. Tegenwoordig is dat vrijwel standaard bij dergelijke apparaten, maar de NMS 1436 is inderdaad al wat ouder.

Afdrukmogelijkheden

De NMS 1461 heeft een hele reeks afdrukmogelijkheden in huis. In de eerste plaats zijn er verschillende 'pitches' of letterbreedtes. Er is keus uit Pica, Elite, Condensed of Proportioneel. In de IBM mode is er zelfs nog een extra kleine Condensed mode beschikbaar waarbij er 20 tekens per inch worden afgedrukt. Daarnaast is er nog een groot aantal stijlen voorhanden. Er is cursief, twee soorten vet – bold en double strike, dubbele breedte en super- en subscript. Daarnaast is er dan nog de mogelijkheid tot onderstrepen.
Er zijn een groot aantal combinaties mogelijk van deze mogelijkheden, in het printvoorbeeld zijn er enkele te zien. Er zijn echter ook een aantal zaken die zich niet laten combineren. Super-en subscript overschrijven bijvoorbeeld cursief en omgekeerd zet cursief super- en subscript opzij.
De meeste lettertypes kunnen ook in Letter Quality worden afgedrukt, maar Condensed en LQ gaan niet samen. Als verschillende modes elkaar uitsluiten geldt er een voorrangsregeling. Zet men LQ aan als er Condensed wordt geprint dan blijft deze oproep even in de wachtkamer, tot de printer in een mode staat waarin hij de Letter-quality wel kan toepassen.

Het frontpaneel

De machine heeft een beige kunststof behuizing en lijkt – uiterlijk althans – exact op zijn voorgangers de VW-0030 en de NMS 1431. Het binnenwerk is goed beschermd tegen stof en vuil. Links voorop de printer treffen we het frontpaneel aan. Het telt vier druktoetsen en vier LED-jes. De toetsen hebben verschillende functies. De On-Line toets spreekt voor zich. On-line staat de printer klaar informatie te ontvangen, Off-Line kan er papier worden ingevoerd of uitgeworpen. De LQ-toets zet de Letter Quality-mode aan of uit.
De NMS 1436 is een 9-naalds matrixprinter. De printer bouwt de lettertekens op door de naaldjes tegen het inktlint te schieten. In de draftmode – de kladstand – zijn de typische matrixpuntjes duidelijk te onderscheiden. In de LQ-mode echter drukt de NMS 1436 een – voor een 9-naalds printer – heel fraai lettertje af. Elke regel wordt in deze stand twee maal afgedrukt; de tweede maal een fractie verschoven, zodat de afzonderlijke stipjes niet meer zijn te onderscheiden.
De Line- en Formfeed toetsen zorgen respectievelijk voor de opvoer van een regel of een bladzij. De On-line toets en de Letter Quality-toets hebben ieder hun eigen signaallampje. De twee andere lichtjes geven Power en Paper Out aan. Door de On Line-toets enige tijd ingedrukt te houden komt men in de pitch-keuze stand. Met de andere toetsen kan dan door de printkop naar een bepaalde positie te dirigeren de gewenste letterbreedte of pitch worden ingesteld. Er is keus uit:

Code: Instelling via een programma;
Pica: 10 tekens per inch;
Elite: 12 tekens per inch;
Condensed Pica: 17 tekens per inch;
Condensed Elite: 20 tekens per inch en
Proportioneel: circa 11 tekens per inch (elk teken heeft eigen breedte).

Welke positie welke type geeft staat op de beschermkap aangegeven. De kleinste breedte, Condensed Elite, is in de MSX mode niet verkrijgbaar. Een via het paneel ingevoerde pitch kan niet meer door de software worden overschreven. Wordt bij het aanzetten de LQ-toets ingedrukt gehouden dan komt de printer in de zelftest-mode, waarbij de machine afwisselend in klad en in Letter Quality de ASCII-codes 32 tot en met 126 uitspuugt.
Het ingedrukt houden van de Formfeed-toets zet de printer in de hexdumpstand. Alle tekens die vanaf dat moment worden gestuurd worden met 16 tekens per regel hexadecimaal weergegeven. Dit kan behulpzaam zijn bij het debuggen van programma's of testen van printerbesturingen.
Wordt de Linefeed bij het inschakelen ingedrukt, dan drukt de machine, naar gelang de basis-configuratie, de instelling van de dipswitches af.

Dipswitches

De NMS 1436 heeft aan de achterzijde twee dipswitch-blokjes, waarmee de basis configuratie – de stand waarin de printer opstart – kan worden ingesteld. De belangrijkste schakelaartjes zorgen voor de mode waarin wordt opgestart: MSX, IBM, Epson of Prestel.
De overige schakelaars hebben – afhankelijk van de gekozen mode – verschillende functies. Er kan onder meer de tekenset en de paginalengte mee worden ingesteld. In MSX-mode kan er uit de MSX- of de Prestel-set worden gekozen. De Prestel-set bestaat uit de characters waarmee videotex-schermen (bijvoorbeeld voor Viditel of ComNet) zijn op te bouwen. In IBM-mode kan er voor set 1 of set 2 worden gekozen. Beide zijn overigens vrijwel identiek. Verder kunnen er verschillende nationale aanpassingen – voor de diacritische tekens in onder meer Frans, Duits en Spaans – op de tekensets worden geselecteerd.

Via de software

De meeste instellingen kunnen softwarematig worden overschreven, zoals de standaard tekenset, maar soms ook niet. Het al of niet met slash weergeven van de nul in MSX mode is enkel via één van de tuimelschakelaars te regelen.
Vanuit de Prestel-mode kunnen de andere modes softwarematig worden aangeroepen. Er hoeven bij het wisselen van mode dus niet telkens dipswitches worden omgezet. Deze schakelaartjes zijn niet gemaakt voor intensief gebruik en gaan bij voortdurend schakelen snel stuk. Het schakelen tussen deze verschillende modes moet echter altijd via de Prestel-mode lopen. Na een resetcommando valt de printer terug in de Prestel-mode en moet de gewenste mode opnieuw worden ingesteld. We durven niet met zekerheid te beweren dat dit vanuit de programmatuur altijd goed gaat. We hadden liever gezien dat deze modes ook blijvend via het frontpaneel konden worden ingesteld.

Op de MSX voldeed de printer prima, maar we hebben de machine natuurlijk ook even aan een PC gehangen. Na het vinden van de juiste driver liet de printer zich goed vanuit WordPerfect besturen. Ook grafieken uit de Harvard Graphics lieten zich heel redelijk afdrukken. Bij het kruisgebruik ging het niet altijd goed. Vanaf de MSX bleek de Epson-tekenset niet softwarematig op te roepen. Andersom bleek vanuit GW-Basic de MSX-mode grafisch niet goed te werken.

Printbuffer

Volgens de specificaties bedraagt de printbuffer in de MSX-mode 8K. Dit betekent dat in er in deze buffer ruim 8000 tekens die naar de printer zijn gestuurd kunnen worden opgeslagen. Ons

<DIP SWITCH SETTINGS>

Table1

Switch		On	Off
1-1	Mode selection		
-2	Mode selection	See Table2	
-3	International character		
-4	International character	See Table3	
-5	No use		
-6	Page length	12"	11"
-7	CR Code selection	CR+LF	AUTO FEED
-8	Download character	Valid	Invalid
2-1	CSF mode	Valid	Invalid
-2	1" Skip perforation	Valid	Invalid

Table2

Mode	Switch	
selection	1-1	1-2
PRESTEL	on	on
EPSON FX	on	off
IBM	off	on
MSX	off	off

Table3

International	Switch	
character	1-3	1-4
U.K.	off	off
SWEDEN	on	off
ITALY	off	on
BELGIUM	on	on

Dip switch settings in MSX-mode

is daar echter weinig van gebleken. We zouden mogen verwachten dat na het versturen van zo'n 5000 tekens de printer na enkele seconden weer vrij is voor de volgende klus, terwijl de printer de buffer verwerkt. De MSX bleek echter pas zo'n 2000 tekens voor het eind van de printopdrachten weer klaar te staan. In de overige modes bedraagt de buffer 2,3K, bij het gebruik van een downloadset 0,7K.

Snelheid

Het mag bekend zijn dat de meeste printerfabrikanten de afdruksnelheid van hun machines schromelijk overdrijven. De opgegeven snelheid wordt slechts gehaald als er met de regelopvoer en het papiertransport geen rekening wordt gehouden.
Om een idee van de effectieve snelheid te krijgen moet men de gespecificeerde snelheid met ten minste een derde verminderen. Bij de NMS 1436 maten wij 80 tekens per seconde in de draft-mode. In de letter-quality stand bereikte de printer 18 characters per seconde.
Een fraaie letter gaat dus ten koste van veel tijd. Ook vetprinten of super- en subscript waarbij er meer puntjes worden gezet of de rol moet worden bewogen kosten meer tijd dan de gewone draft-mode.

Tenslotte

Met de NMS 1436 heeft de PTC een interessante aanbieding in huis. De printer is zowel op een MSX als op een PC direct inzetbaar. Het is echter jammer dat de extra mogelijkheden van de IBM- of Epson-stand in de MSX-mode niet kunnen worden benut.
De bijgeleverde documentatie is voldoende. Het installeren van de printer wordt in zes talen beschreven. Het overzicht van de besturingscodes en hun betekenis is er echter alleen in het Engels.

De beschrijving van de codes is in twee gedeeltes uitgesplitst. Eén deel voor de MSX-mode en één voor de overige modes. Het geheel is overzichtelijk en ruim van voorbeelden voorzien. Een tractorfeed wordt standaard bij de printer geleverd. Voor de kabels zult u zelf moeten zorgen, hetgeen voor de MSX printerkabel even zoeken kan zijn.

Zoals al gezegd, de NMS 1436 is officieel al een hele tijd uit productie. Dat deze printer nu toch weer aangeboden wordt is een gelukje. Blijkbaar had men ergens in een magazijn nog een stapel dozen staan.
Philips doet niet meer aan MSX, dus Philips kan zelf weinig meer met deze printers aanvangen. Vandaar dat de NMS 1436 via de PTC, de Philips Thuiscomputer Club wordt verkocht. Een soort gebruikersgroep die zich geheel op Philips-machines richt.

Meestal kunnen niet-leden de aanbiedingen van de PTC ook betrekken, zij het tegen een hogere prijs, maar in dit geval verlangt de PTC wel dat u lid bent. Wie de printer graag wil hebben zal dus tegelijkertijd lid moeten worden. Leden kunnen van alle aanbiedingen profiteren en krijgen een abonnement op het PTC-blad PRINT.

Product: NMS 1436 Multicompatible printer
Prijs: *f* 545,-
Transportkosten: *f* 25,-
Het verplichte PTC-lidmaatschap kost *f* 46,- per jaar.

Bestellen en nadere informatie:

Tel.: 040 - 758912
Bellen op werkdagen tussen 10.00 en 12.00 of van 13.00 tot 16.00 uur.

Listing: Z80 assembler

Zoals iedere MSX programmeur weet blinken Basic programma's niet echt uit in snelheid. Om toch een snel programma te krijgen is meestal een dosis machinetaal nodig – hetgeen resulteert in de hexadecimale getallenbrei die zo vaak in dit blad is te bewonderen. Nu is het algemeen bekend onder de ML fanatici dat voor het grotere ML werk een assembler onontbeerlijk is. Helaas zijn de meeste assemblers nogal prijzig. Daarom presenteren wij u bij deze, met gepast trots, ZAS. Dat staat voor Z80 ASsembler, een eenvoudige assembler in Basic.

ASSEMBLER VOOR MSX IN BASIC

ZAS is bedoeld voor MSX computers met één of meer diskdrives. Het assembler programma staat in een aparte file die als invoer dient, terwijl het machinetaal programma naar een uitvoerfile geschreven wordt. Dat machinetaal programma kan gewoon vanuit Basic met de 'BLOAD' opdracht geladen en gestart worden. Het assembler programma zelf kan met de gewone Basic editor gemaakt worden. Omdat Basic niets aankan met de assembler opdrachten wordt alle assembler-taal in REM regels gezet.

Voorbeeld

Een voorbeeld staat in listing 1. Dit programmaatje drukt een deel van de MSX karakterset af op het scherm. Merk op dat in plaats van de REM opdracht steeds het ' tekentje gebruikt wordt. Verder moet het assembler programma altijd als een tekst bestand worden gesaved, omdat de assembler alleen tekstfiles kan verwerken. Gebruik bij het saven dus bijvoorbeeld:

SAVE "VB.ASM",A

Om een eenmaal gesaved assembler programma te assembleren dient ZAS gedraaid te worden. Na gestart te zijn laadt ZAS eerst enkele interne tabellen, waarna gevraagd wordt om de invoerfile. Wanneer nu een naam zonder extensie wordt opgegeven plakt ZAS automatisch de extensie '.ASM' achter de opgegeven naam. Vervolgens wordt gevraagd om de naam van de list file. Ook hier wordt eventueel de naam aangevuld met een extensie, alleen wordt hier '.LST' toegevoegd.
Wanneer voor de speciale naam 'nul' – dus zonder punt en extensie erachter – wordt gekozen, dan wordt helemaal geen list file gemaakt. Die list file overigens geeft aan hoe het programma is geassembleerd en welke waarden zijn toegekend aan de diverse labels. Hierna gaat ZAS aan de slag.

Eventuele fouten worden niet alleen op het scherm maar ook in een speciale error file gerapporteerd. Die error file heeft dezelfde naam als de invoerfile, alleen is de extensie hier '.ERR'. Wanneer geen fouten worden gevonden wordt het machinetaal programma weggeschreven naar een file die ook dezelfde naam heeft als het assemblerprogramma. De extensie is echter '.BIN'.

Zoals gezegd kan het machinetaalprogramma gewoon vanuit Basic gedraaid worden, bijvoorbeeld met:

BLOAD "VB.BIN",R

De error file en list file zien er net als het assembler programma uit als Basic programma's, die met de LOAD opdracht kunnen worden geladen. Het machinetaal programma kan ook binnen een Basic programma gebruikt worden. Een voorbeeld staat in listing 2.

Syntax

Bij het schrijven van het assembler programma gelden de volgende regeltjes: Labels, instructies en dergelijke kunnen zowel in hoofdletters als in kleine letters geschreven worden. De assembler trekt zich daar niets van aan, het programma is dus niet 'case sensitive'. Dit houdt overigens wel in dat de assembler geen verschil ziet tussen bijvoorbeeld 'LABEL' en 'label'.

Labels staan aan het begin van de regel en worden gevolgd door een dubbele punt. Door de dubbele punt weet de assembler dat het om een label gaat en niet om een instructie.
Overigens is het best toegestaan om spaties voor het label te zetten. Een label mag echter niet beginnen met een karakter dat geen letter is.
Alles wat achter een puntkomma staat wordt opgevat als commentaar; hexadecimale getallen dienen te worden vooraf gegaan door een $ teken.

Pseudo-ops

Pseudo-ops zijn instructies voor de assembler zelf, ze worden dus niet geassembleerd tot Z80 instructies. De ZAS programmeur kan beschikken over het volgende rijtje:

EQU

Met EQU kan een constante gedefinieerd worden, zoals in:

10 ' equ naam, nn

```
10 ' org $cØØØ ; ml moet vanaf CØØØ        Ø
20 '            ; geladen worden            Ø
30 ' equ chput, $ØØa2                       Ø
40 ' ld b, 223 ; 223 karakters              Ø
50 'volgende:                               Ø
60 ' ld a,b                                 Ø
70 ' add a, 32  ; ASCII codes lopen v
anaf 32                                     Ø
80 ' push bc                                Ø
90 ' call chput ; zet kar in A op sch
erm                                         Ø
100 ' pop bc                                Ø
110 ' djnz volgende                         Ø
120 ' ret ; terug naar BASIC                Ø
```

Listing 1, VB.ASM

```
1Ø ' reserveer geheugenruimte voor ml       Ø
2Ø CLEAR 1ØØ,&HCØØØ                       161
3Ø ' laadt ml (vanaf &hcØØØ, vanwege        Ø
4Ø ' org $cØØØ in assembler programma
)                                           Ø
5Ø BLOAD "vb.bin"                          83
6Ø ' geef op dat ml routine Ø vanaf         Ø
7Ø ' &hcØØØ in geheugen staat               Ø
8Ø DEF USRØ=&HCØØØ                        243
9Ø PRINT "De karakterset van de MSX i
s:"                                        4Ø
1ØØ ' roep ml routine Ø aan                 Ø
11Ø I=USRØ(Ø)                              56
```

Listing 2, VB.BAS

waarbij nn een getal is en naam de naam van de constante. De constante kan dan verder in plaats van het getal worden gebruikt, behalve in andere EQU opdrachten. De constructie:

10 ' equ naam1, 5
20 ' equ naam2, naam1

is dus niet toegestaan.

Overigens wordt een label definitie door ZAS intern net zo afgehandeld als een constante definitie met EQU. Zo is het toegestaan om te schrijven:

10 ' equ adresbuf, $d000
20 ' label:
30 ' ld hl, label
40 ' ld (adresbuf),hl

ORG

Met de opdracht:

10 ' org nn

wordt het beginadres van het machinetaal programma opgegeven. BLOAD zal het programma vanaf dit adres in het geheugen laden. Wanneer een programma geen ORG opdracht heeft wordt als beginadres &hC800 genomen. ORG mag hoogstens één keer in het programma voorkomen en moet dan als eerste opdracht worden gebruikt!

DB

Deze opdracht wordt gebruikt om een enkel byte in het geheugen te zetten. Bijvoorbeeld: 40 ' db 100

DW

Dit is de grote broer van DB. In plaats van een byte zet deze een woord in het geheugen. Bijvoorbeeld:

40 ' dw $c800

Natuurlijk werkt DW zoals we mogen verwachten, eerst de low-byte en dan de high-byte. Wie zich nu afvraagt waar we het in hemelsnaam over hebben: zie de eerste aflevering van de nieuwe ML-cursus in het volgende nummer.

Gebruiksgemak

Natuurlijk is ZAS niet zo snel als één van de bestaande commerciële assemblerprogramma's.
We zeiden al in de inleiding dat Basic-

VB.LST uitvoer van VB.ASM

```
1  '  C000              10   org $c000 ; ml moet vanaf C000
2  '  C000              20             ; geladen worden
3  '  C000              30   equ chput, $00a2
4  '  C000  06 DF       40   ld b, 223 ; 223 karakters
5  '  C002              50  volgende:
6  '  C002  78          60   ld a,b
7  '  C003  C6 20       70   add a, 32  ; ASCII codes lopen vanaf 32
8  '  C005  C5          80   push bc
9  '  C006  CD A2 00    90   call chput ; zet kar in A op scherm
10  '  C009  C1          100   pop bc
11  '  C00A  10 F6       110   djnz volgende
12  '  C00C  C9          120   ret ; terug naar BASIC
13  '
14  '
15  '  dec      hex    label
16  '  -------  ----   -----
17  '      162    A2   CHPUT
18  '   -16382  C002   VOLGENDE
```

programma's niet zo rap zijn.
ZAS biedt echter wel heel wat gebruiksgemak. Omdat de assembler gewoon onder Basic kan worden opgestart, dezelfde omgeving waar we de ML-source in ontwikkelen, hoeft men niet voortdurend editors en dergelijke te laden. Temeer daar ook de uiteindelijke ML – het .BIN bestand – gewoon onder Basic getest kan worden. Qua mogelijkheden met list- en error-bestanden is ZAS behoorlijk compleet.

ML-cursus

Met ZAS als assembler gaan we, in de komende nummers, meer aandacht aan machinetaal-programmering besteden. Rond ZAS zal een cursus machinetaal verschijnen, met kleine, aardige voorbeeldjes van de snelheid die er in ML kan worden bereikt.

Uiteindelijk is en blijft nu eenmaal het feit dat men alleen onder ML de hele MSX werkelijk kan beheersen. Alles wat ASCII en MicroSoft verboden hebben is onder ML te realiseren, doordat men in machinetaal ook regelrecht in de chips kan ingrijpen. En dat is, naast de snelheid, een heel goede reden om ML te beheersen, als men echt het onderste uit de MSX wil halen.

Z80 hoofdlisting

```
10 REM ZAS - Z80 ASsembler                   0
20 REM                                       0
30 REM MSX Computer Magazine                 0
40 REM                                       0
50 REM initialisatie                         0
60 MAXFILES=4: CLEAR 1500: DEFINT A-Z
: BS=&HC800: NE=0: NL=0: NIL=256: TRU
E=(1=1): FALSE=(1=2): OF=0: LN=1: FB=
TRUE: EN=1                                  86
70 KEY 1,"load"+CHR$(34)+"tst.asm"+CH
R$(34)                                      62
80 KEY 2,"list "                           162
90 KEY 3,"load"+CHR$(34)+"tst.err"+CH
R$(34)                                      81
100 KEY 4,"save"+CHR$(34)+"tst.asm"+C
HR$(34)+",a"                               146
110 KEY 5,"load"+CHR$(34)+"zas.bas"+C
HR$(34)+",r"                                20
120 KEY 6,"load"+CHR$(34)+"tst.lst"+C
HR$(34)                                    178
130 KEY 9,"bload"+CHR$(34)+"tst.bin"+
CHR$(34)+",r"                              122
140 KEY 10,"save"+CHR$(34)+"zas.bas"+
CHR$(34)                                    16
150 REM codebuffer                           0
160 SA=BS: CS=BS                           219
170 REM labelopslag                          0
180 DIM LH(25),LN$(255),LX(255),LV(25
5): FOR I=0 TO 25: LH(I)=NIL: NEXT         192
190 REM opslag mnemonics                     0
200 DIM MN$(24),SI(24),GR(71),CO(71):
 H$="?": I=0: J=-1: RESTORE 930            115
210 READ A$: IF A$="*" THEN MN$(J)=H$
: GOTO 290                                 211
220 READ G,C$: K=ASC(A$)-ASC("A"): A$
=RIGHT$(A$+STRING$(4-LEN(A$),"."),3)       190
230 GR(I)=G: CO(I)=VAL("&H"+C$): I=I+
1                                           57
240 IF K=J THEN H$=H$+A$: GOTO 210          66
250 IF J<>-1 THEN MN$(J)=H$                160
260 J=K: SI(J)=I-1: H$=A$: GOTO 210         43
270 REM --hoofdprogramma                     0
280 REM vraag filenamen                      0
290 SF$="tst": PRINT "Sourcefilenaam
(default = ";SF$;")";: INPUT SF$: I=I
NSTR(SF$,"."): IF I=0 THEN EF$=SF$+".
err": BF$=SF$+".bin": SF$=SF$+".asm"
ELSE EF$=LEFT$(SF$,I)+"err": BF$=LEFT
$(SF$,I)+"bin"                             100
300 LF$="NUL": PRINT "Listfilenaam (d
efault = ";LF$;")";: INPUT LF$: IF (I
NSTR(LF$,".")=0) AND (INSTR(LF$,":")=
0) THEN LF$=LF$+".lst"                     208
310 PASS=1: OPEN LF$ FOR OUTPUT AS #2
: OPEN BF$ FOR OUTPUT AS #3: OPEN EF$
 FOR OUTPUT AS #4                           83
320 REM open files                           0
330 BP=BS: IR$="": RN$="": NB=0: ON E
RROR GOTO 1060: OPEN SF$ FOR INPUT AS
 #1: ON ERROR GOTO 0                        73
340 REM lees volgende regel                  0
350 IF PASS=2 THEN FOR I=NB TO 5: PRI
NT #2,"   ";: NEXT: PRINT #2,RN$;" ";
IR$                                         94
360 NB=0                                   236
370 IF EOF(1) THEN GOTO 510                146
380 LINE INPUT #1,IR$: IF ASC(LEFT$(I
R$,1))=255 THEN ER$=SF$+" niet gesave
d met ,a optie": GOSUB 1350: GOTO 580
 ELSE IP=1: I=INSTR(IR$,"'"): IF I=0
THEN RN$="" ELSE RN$=LEFT$(IR$,I-1):
IR$=RIGHT$(IR$,LEN(IR$)-I)                 131
390 PRINT USING "\   \";RN$;: PRINT C
HR$(29);CHR$(29);CHR$(29);CHR$(29);CH
R$(29);                                     13
400 IF PASS=2 THEN PRINT #2,LN;"  '  "
;HEX$(CS+BP-BS);" ";: LN=LN+1               37
410 REM assembleer regel                     0
420 GOSUB 1080: IF K$=";" THEN GOTO 3
50                                         154
430 GOSUB 1180: IF K$=": " THEN W=CS+
BP-BS: GOSUB 1240: IP=IP+1: GOSUB 108
```

```
Ø: IF K$=";" THEN GOTO 35Ø ELSE GOSUB
 118Ø 'labeldef                              81
44Ø REM verwerk mnemonic in NA$              Ø
45Ø IF LEN(NA$)>4 OR LEN(NA$)<=Ø THEN
 GOTO 49Ø                                    25
46Ø  J=ASC(NA$): IF J>ASC("Z") THEN G
OTO 49Ø                                      54
47Ø  J=J-ASC("A"): I=INSTR(MN$(J),RIG
HT$(NA$+STRING$(4-LEN(NA$),"."),3)):
IF I>Ø THEN I=SI(J)+(I-1)/3: AW=GR(I)
: AI=CO(I): GOSUB 14ØØ: GOSUB 127Ø: G
OTO 35Ø                                      77
48Ø REM foute mnemonic                       Ø
49Ø IF NA$="" THEN ER$="Geen label of
 mnemonic gevonden" ELSE ER$=NA$+" is
 geen mnemonic"                              155
5ØØ GOSUB 135Ø: GOTO 35Ø                     137
51Ø IF NE<>Ø THEN GOTO 57Ø ELSE IF PA
SS=1 THEN PASS=PASS+1: CE=BP+(CS-BS):
 CLOSE #1: GOTO 33Ø                          254
52Ø REM geef labelbindings                   Ø
53Ø IF NL=Ø THEN GOTO 57Ø                    42
54Ø RN$="": PRINT #2,LN;" ' ": PRINT
#2,LN+1;" ' ": PRINT #2,LN+2;" ' dec
     hex   label": PRINT #2,LN+3;" '
-------  ----  -----"                        2Ø9
55Ø FOR I=Ø TO NL-1: PRINT #2,LN+4+I;
" ' ";: PRINT #2,USING "#######";LV(I
);: PRINT #2,"  ";STRING$(4-LEN(HEX$(
LV(I)))," ");HEX$(LV(I));"  ";LN$(I):
 NEXT                                        185
56Ø REM sluit alle files en eindig           Ø
57Ø IF NE>Ø THEN PRINT "Aantal fouten
: ";NE                                       7
58Ø CLOSE: ON ERROR GOTO 61Ø                 97
59Ø IF NE=Ø THEN KILL EF$ ELSE KILL B
F$: KILL LF$                                 226
6ØØ END                                      175
61Ø RESUME NEXT                              121
62Ø REM -- emitbyte                          Ø
63Ø IF PASS=1 THEN GOTO 66Ø                  85
64Ø  IF FB THEN FB=FALSE: PRINT #3,CH
R$(254);: J=CS: GOSUB 71Ø: J=CE: GOSU
B 71Ø: J=SA: GOSUB 71Ø                       198
65Ø  PRINT #3,CHR$(B);: PRINT #2," ";
: A$=HEX$(B): IF LEN(A$)=1 THEN PRINT
 #2,"Ø";A$; ELSE PRINT #2,A$;                155
66Ø BP=BP+1: NB=NB+1: RETURN                 173
67Ø REM -- emitword                          Ø
68Ø K$=STRING$(4-LEN(HEX$(B)),"Ø")+HE
X$(B)                                        18Ø
69Ø B=VAL("&H"+RIGHT$(K$,2)): GOSUB 6
3Ø: B=VAL("&H"+LEFT$(K$,2)): GOTO 63Ø        7Ø
7ØØ REM -- emitword zonder naar listf
ile te gaan                                  Ø
71Ø K$=STRING$(4-LEN(HEX$(J)),"Ø")+HE
X$(J)                                        199
72Ø PRINT #3,CHR$(VAL("&H"+RIGHT$(K$,
2)));CHR$(VAL("&H"+LEFT$(K$,2)));: RE
TURN                                         47
73Ø REM -- labval                            Ø
74Ø IF PASS=1 THEN AW=Ø: RETURN              253
75Ø I=LH(ASC(NA$)-ASC("A")): IF I=NIL
 THEN FK=5: RETURN                           2Ø4
76Ø IF LN$(I)=NA$ THEN AW=LV(I): RETU
RN ELSE I=LX(I): IF I=NIL THEN FK=5:
RETURN ELSE GOTO 76Ø                         1Ø4
77Ø REM -- scan                              Ø
78Ø GOSUB 1Ø8Ø: IF K$=";" THEN TK$="G
": RETURN                                    1Ø5
79Ø IF K$="(" THEN TK$="B": GOTO 91Ø         2Ø9
8ØØ IF K$=")" THEN TK$="E": GOTO 91Ø         23
81Ø IF K$="," THEN TK$="K": GOTO 91Ø         211
82Ø IF K$="$" THEN IP=IP+1: GOSUB 112
Ø: TK$="C": AW=VAL("&H"+NA$): RETURN         93
83Ø GOSUB 118Ø: IF LEN(NA$)=Ø THEN ER
$="onverwacht karakter '"+K$+"' aange
troffen": TK$="G": GOTO 135Ø ELSE IF
LEN(NA$)=1 THEN NA$=NA$+" "                  23
84Ø I=INSTR("B C D E H L   A R I",NA$
): IF I<>Ø THEN TK$="H": AW=(I-1)/2:
RETURN                                       12
85Ø IF K$="+" THEN IP=IP+1: TK$="I":
IF NA$="IX" THEN AW=&HDD: RETURN ELSE
 IF NA$="IY" THEN AW=&HFD: RETURN ELS
E ER$="Geen IX of IY voor +": GOTO 13
5Ø                                           153
86Ø I=INSTR("NZZ NC  POPEP M",NA$): I
F I<>Ø THEN TK$="J": AW=(I-1)/2: RETU
RN                                           179
87Ø I=INSTR("BCDEHLSPIXIY  AF",NA$):
IF I<>Ø THEN TK$="D": AW=(I-1)/2: RET
URN                                          2ØØ
88Ø K$=LEFT$(NA$,1): IF (K$>="Ø") AND
 (K$<="9") THEN TK$="C": AW=VAL(NA$):
 RETURN                                      118
89Ø TK$="L": RETURN                          246
9ØØ REM werk (,) en , verder af              Ø
91Ø IP=IP+1: RETURN                          214
92Ø REM --- data                             Ø
93Ø DATA ADC,2,88,ADD,3,8Ø,AND,4,AØ,B
IT,5,4Ø,CALL,6,C4,CCF,1,3F,CP,4,B8           2ØØ
94Ø DATA CPD,1,A9ED,CPDR,1,89ED,CPI,1
,A1ED,CPIR,1,B1ED,CPL,1,2F,DAA,1,27          47
95Ø DATA DB,Ø,1,DEC,7,5,DI,1,F3,DJNZ,
18,1Ø,DW,Ø,2,EI,1,FB,END,Ø,5,EQU,Ø,3         238
96Ø DATA EX,8,Ø,EXX,1,D9,HALT,1,76           128
97Ø DATA IM,9,Ø,IN,1Ø,DB,INC,7,4,IND,
1,AAED,INDR,1,BAED,INI,1,A2ED                238
98Ø DATA INIR,1,B2ED,JP,11,C2,JR,12,2
Ø,LD,13,4Ø,LDD,1,A8ED,LDDR,1,B8ED,LDI
,1,AØED                                      176
99Ø DATA LDIR,1,BØED,NEG,1,44ED,NOP,1
,Ø,OR,4,BØ,ORG,Ø,4,OTDR,1,8BED               11
```

```
1000 DATA OTIR,1,B3ED,OUT,19,D3,OUTD,
1,ABED,OUTI,1,A3ED,POP,14,C1,PUSH,14,
C5                                        24
1010 DATA RES,5,80,RET,15,C0,RETI,1,4
DED,RETN,1,45ED,RL,16,10,RLA,1,17          8
1020 DATA RLC,16,0,RLCA,1,7,RLD,1,6FE
D,RR,16,18,RRA,1,1F,RRC,16,8,RRCA,1,F    181
1030 DATA RRD,1,67ED,RST,17,C7,SBC,2,
98,SCF,1,37,SET,5,C0,SLA,16,20           170
1040 DATA SRA,16,28,SRL,16,38,SUB,4,9
0,XOR,4,A8                               139
1050 DATA *                              230
1060 PRINT "*** ";SF$;" kan niet geop
end worden.": END                        206
1070 REM -- skip spaties                   0
1080 IF IP>LEN(IR$) THEN K$=";" ELSE
K$=MID$(IR$,IP,1): IF ASC(K$)<=32 THE
N IP=IP+1: GOTO 1080                     184
1090 IF K$>="a" AND K$<="z" THEN K$=C
HR$(ASC(K$)-ASC("a")+ASC("A"))           123
1100 RETURN                              125
1110 REM -- lees hexadecimale string       0
1120 I=IP                                  2
1130 IF IP<=LEN(IR$) THEN K$=MID$(IR$
,IP,1): IF ((K$>="a") AND (K$<="f"))
OR ((K$>="A") AND (K$<="F")) OR ((K$>
="0") AND (K$<="9")) THEN IP=IP+1: GO
TO 1130                                  110
1140 IF IP<=LEN(IR$) AND K$<>" " AND
K$<>"," AND K$<>";" AND K$<>")" THEN
ER$="'"+K$+"' gevonden in hexadecimaa
l getal": GOTO 1350                       92
1150 NA$=MID$(IR$,I,IP-I)                 74
1160 RETURN                              143
1170 REM -- lees string                    0
1180 I=IP                                 20
1190 IF IP<=LEN(IR$) THEN K$=MID$(IR$
,IP,1): IF ((K$>="a") AND (K$<="z"))
OR ((K$>="A") AND (K$<="Z")) OR ((K$>
="0") AND (K$<="9")) THEN IP=IP+1: GO
TO 1190                                   80
1200 NA$=MID$(IR$,I,IP-I): FOR J=1 TO
 IP-I                                     96
1210 IF MID$(NA$,J,1)>="a" AND MID$(N
A$,J,1)<="z" THEN MID$(NA$,J,1)=CHR$(
ASC(MID$(NA$,J,1))-ASC("a")+ASC("A"))    183
1220 NEXT: RETURN                        212
1230 REM -- deflabel                       0
1240 IF PASS=1 THEN IF ASC(NA$)>=ASC(
"A") AND ASC(NA$)<= ASC("Z") THEN I=A
SC(NA$)-ASC("A"): LN$(NL)=NA$: LV(NL)
=W: LX(NL)=LH(I): LH(I)=NL: NL=NL+1 E
LSE ER$=NA$+" is geen goed label": GO
TO 1350                                   69
1250 RETURN                              142
1260 REM -- foutm                          0
1270 IF FK=0 THEN RETURN                   5
1280 ON FK GOTO 1290,1300,1310,1320,1
330                                      255
1290 ER$="Fout in operand": GOTO 1350    217
1300 ER$="Getal te groot of te klein"
: GOTO 1350                              129
1310 ER$="Komma verwacht": GOTO 1350     199
1320 ER$="Doeladres buiten bereik rel
atieve sprong": GOTO 1350                131
1330 ER$="Label "+NA$+" niet gedefini
eerd": GOTO 1350                         191
1340 REM -- err                            0
1350 PRINT #4,EN;" ' *** ";: IF RN$<>
"" THEN PRINT #4," Regel ";RN$;": ";     148
1360 PRINT #4,ER$: EN=EN+1               208
1370 PRINT "*** ";: IF RN$<>"" THEN P
RINT "Regel ";RN$;": ";                   86
1380 PRINT ER$: PRINT: NE=NE+1: RETUR
N                                        200
1390 REM --instruct                        0
1400 OC=AI: FK=0: ON AW+1 GOTO 1420,1
520,1550,1650,1750,1820,1900,1940,206
0,2170,2260,2340,2430,2470,2890,2940,
2960,3020,3050,3070                       60
1410 REM pseudo-ops                        0
1420 ON AI GOTO 1430,1440,1450,1490,1
500                                       79
1430 GOSUB 780: GOSUB 3160: IF (NOT N
) OR W THEN GOTO 3370 ELSE B=AW: GOSU
B 630: GOTO 780                          183
1440 GOSUB 780: GOSUB 3160: IF (NOT N
) THEN GOTO 3370 ELSE B=AW: GOSUB 680
: GOTO 780                                14
1450 GOSUB 780: IF TK$<>"L" THEN GOTO
 3370                                    127
1460 BN$=NA$: GOSUB 780: IF TK$<>"K"
THEN GOTO 3370                           133
1470 GOSUB 780: IF TK$<>"C" THEN GOTO
 3370 ELSE GOSUB 3160: IF NOT N THEN
GOTO 3370                                 43
1480 W=AW: NA$=BN$: GOSUB 1240: GOTO
780                                       45
1490 GOSUB 780: GOSUB 3160: IF NOT N
THEN GOTO 3370 ELSE CS=AW: SA=AW: GOT
O 780                                     89
1500 GOSUB 780: GOSUB 3160: IF NOT N
THEN GOTO 3370 ELSE SA=AW: GOTO 780      179
1510 REM groep 1 (instructies met imp
liciete adressering)                       0
1520 B=AI: IF (AI AND &HFF00) = 0 THE
N GOSUB 630 ELSE GOSUB 680               182
1530 GOTO 780                            123
1540 REM groep 2 (adc, sbc)                0
1550 GOSUB 780                             2
1560 IS=(TK$="H") AND (AW=7)              72
1570 IF NOT IS THEN IF (TK$<>"D") OR
(AW<>2) THEN GOTO 3370                    37
1580 GOSUB 780: IF TK$<>"K" THEN FK=3
```

```
: RETURN                                      132
159Ø IF IS THEN GOTO 175Ø                     129
16ØØ GOSUB 78Ø: IF (TK$<>"D") OR (AW<
Ø) OR (AW>3) THEN GOTO 337Ø                    92
161Ø B=&HED: GOSUB 63Ø                         88
162Ø IF OC=&H88 THEN OC=&H4A ELSE OC=
&H42                                          214
163Ø GOTO 342Ø                                219
164Ø REM groep 3 (add)                          Ø
165Ø GOSUB 78Ø: IF TK$="H" THEN IF AW
<>7 THEN GOTO 337Ø ELSE GOTO 173Ø 'ad
d a,                                          241
166Ø IF (TK$<>"D") OR ((AW<>2) AND (A
W<>4) AND (AW<>5)) THEN GOTO 337Ø             242
167Ø IF (AW=4) OR (AW=5) THEN GOSUB 3
44Ø                                            26
168Ø GOSUB 78Ø: IF TK$<>"K" THEN GOTO
 337Ø                                         114
169Ø ID=AW: GOSUB 78Ø: IF (TK$<>"D")
OR NOT ((AW<2) OR (AW=3) OR (AW=ID))
THEN GOTO 337Ø                                 12
17ØØ IF AW=ID THEN AW=2                       237
171Ø OC=9                                      42
172Ø GOTO 342Ø                                218
173Ø GOSUB 78Ø: IF TK$<>"K" THEN GOTO
 337Ø                                         1Ø1
174Ø REM groep 4 (and,cp,or,sub,xor)
en adc a, en sbc a, en add a, en ld r
, waarbij r<>"a"                                Ø
175Ø GOSUB 78Ø: GOSUB 321Ø: IF FK<>Ø
THEN RETURN                                    26
176Ø IF AO>3 THEN GOTO 337Ø                    68
177Ø ON AO GOTO 178Ø,179Ø,18ØØ                148
178Ø B=OC+AV: GOTO 63Ø                          6
179Ø B=OC XOR &H46: GOSUB 63Ø: IF (AN
 AND &HFFØØ)<>Ø THEN FK=2: RETURN ELS
E B=AN: GOTO 63Ø                              239
18ØØ IX=OC+6: GOTO 34ØØ                        9Ø
181Ø REM groep 5 (bit,set,res)                  Ø
182Ø GOSUB 78Ø: GOSUB 316Ø: IF NOT N
THEN GOTO 337Ø ELSE IF (AW>7) OR (AW<
Ø) THEN FK=2: RETURN                           6Ø
183Ø OC=OC+8*AW                               216
184Ø GOSUB 78Ø: IF TK$<>"K" THEN FK=3
: RETURN                                      126
185Ø GOSUB 78Ø: GOSUB 321Ø: IF (FK<>Ø
) OR (AO>3) THEN GOTO 337Ø                    153
186Ø ON AO GOTO 187Ø,337Ø,188Ø                 26
187Ø B=&HCB: GOSUB 63Ø: B=OC+AV: GOTO
 63Ø                                           38
188Ø IX=OC+6: GOTO 348Ø                        34
189Ø REM groep 6 (call)                         Ø
19ØØ GOSUB 35ØØ                               116
191Ø IF FK=1 THEN FK=Ø: B=&HCD: GOSUB
 63Ø ELSE GOSUB 78Ø: IF TK$<>"K" THEN
 FK=3: RETURN ELSE GOSUB 78Ø                  212
192Ø GOTO 355Ø                                 17
193Ø REM groep 7 (dec,inc)                      Ø
194Ø GOSUB 78Ø: IF TK$="D" THEN GOTO
2Ø1Ø                                           85
195Ø GOSUB 321Ø: IF FK<>Ø THEN RETURN         157
196Ø IF AO>3 THEN GOTO 337Ø                    72
197Ø ON AO GOTO 198Ø,337Ø,199Ø                125
198Ø B=OC+8*AV: GOTO 63Ø                       48
199Ø IX=OC+&H3Ø: GOTO 34ØØ                    1Ø3
2ØØØ REM rr                                     Ø
2Ø1Ø IF OC=5 THEN OC=11 ELSE OC=3             126
2Ø2Ø IF AW<4 THEN GOTO 342Ø 'bc,de,hl
,sp                                             7
2Ø3Ø IF AW=7 THEN GOTO 337Ø 'af                7Ø
2Ø4Ø GOSUB 344Ø: B=OC+&H2Ø: GOSUB 63Ø
: GOTO 78Ø 'ix,iy                              13
2Ø5Ø REM groep 8 (EX)                           Ø
2Ø6Ø GOSUB 78Ø: IF TK$<>"D" THEN GOTO
 2Ø9Ø 'ex (sp),..                              91
2Ø7Ø IF AW=7 THEN GOSUB 78Ø: IF TK$<>
"K" THEN FK=3: RETURN ELSE GOSUB 78Ø:
 IF (TK$<>"D") OR (AW<>7) THEN GOTO 3
37Ø ELSE B=8: GOSUB 63Ø: GOTO 78Ø             177
2Ø8Ø IF AW=1 THEN GOSUB 78Ø: IF TK$<>
"K" THEN FK=3: RETURN ELSE GOSUB 78Ø:
 IF (TK$<>"D") OR (AW<>2) THEN GOTO 3
37Ø ELSE B=&HEB: GOSUB 63Ø: GOTO 78Ø          152
2Ø9Ø IF TK$<>"B" THEN GOTO 337Ø               1Ø1
21ØØ GOSUB 78Ø: IF (TK$<>"D") OR (AW<
>3) THEN GOTO 337Ø                            227
211Ø GOSUB 78Ø: IF TK$<>"E" THEN GOTO
 337Ø                                         184
212Ø GOSUB 78Ø: IF TK$<>"K" THEN FK=3
: RETURN                                      1Ø7
213Ø GOSUB 78Ø: IF (TK$<>"D") OR ((AW
<>2) AND (AW<>4) AND (AW<>5)) THEN GO
TO 337Ø                                       192
214Ø IF (AW=4) OR (AW=5) THEN GOSUB 3
44Ø                                             8
215Ø B=&HE3: GOSUB 63Ø: GOTO 78Ø               37
216Ø REM groep 9 (im)                           Ø
217Ø GOSUB 78Ø: GOSUB 316Ø: IF NOT N
THEN GOTO 337Ø                                234
218Ø IF (AW>2) OR (AW<Ø) THEN FK=2: R
ETURN                                         225
219Ø B=&HED: GOSUB 63Ø                        1Ø3
22ØØ ON AW+1 GOSUB 222Ø,223Ø,224Ø              61
221Ø GOSUB 63Ø: GOTO 78Ø                      153
222Ø B=&H46: RETURN                            39
223Ø B=&H56: RETURN                            52
224Ø B=&H5E: RETURN                           22Ø
225Ø REM groep 1Ø (in)                          Ø
226Ø GOSUB 78Ø: IF (TK$<>"H") THEN GO
TO 337Ø                                       2Ø7
227Ø REG=AW: GOSUB 78Ø: IF TK$<>"K" T
HEN FK=3: RETURN                               55
228Ø GOSUB 78Ø: IF TK$<>"B" THEN GOTO
 337Ø                                         129
```

```
2290 GOSUB 780: IF (TK$="H") AND (AW=
1) THEN B=&HED: GOSUB 630: B=&H40+8*R
EG: GOSUB 630: GOTO 3350 'in r,(c)          156
2300 GOSUB 3160: IF NOT N OR (REG<>7)
 THEN GOTO 3370                              71
2310 IF W THEN FK=2: RETURN                 152
2320 B=&HDB: GOSUB 630: B=AW: GOSUB 6
30: GOTO 3350                                13
2330 REM groep 11 (jp)                        0
2340 GOSUB 3500                             117
2350 IF FK=0 THEN GOSUB 780: IF TK$<>
"K" THEN FK=3: RETURN ELSE GOSUB 780:
 GOTO 3550                                  217
2360 FK=0                                   227
2370 IF TK$<>"B" THEN B=&HC3: GOSUB 6
30: GOTO 3550                               180
2380 REM (hl), (ix), (iy)                     0
2390 GOSUB 780: IF (TK$<>"D") OR ((AW
<>2) AND (AW<>4) AND (AW<>5)) THEN GO
TO 3370                                     214
2400 IF (AW=4) OR (AW=5) THEN GOSUB 3
440                                           2
2410 B=&HE9: GOSUB 630: GOTO 3350            30
2420 REM groep 12 (jr)                        0
2430 GOSUB 3500                             116
2440 IF FK=1 THEN FK=0: B=&H18: GOSUB
 630 ELSE IF AW>3 THEN GOTO 3370 ELSE
 GOSUB 780: IF TK$<>"K" THEN FK=3: RE
TURN ELSE GOSUB 780                          95
2450 GOTO 3590                               69
2460 REM groep 13 (ld)                        0
2470 GOSUB 780: GOSUB 3210: IF FK<>0
THEN RETURN                                  27
2480 IF TK$<>"K" THEN FK=3: RETURN           30
2490 ON AO GOTO 2500,3370,2590,2630,2
760,2810,2820                               144
2500 IF AV<6 THEN OC=&H40+8*AV: GOTO
1750 'ld r,.., waarbij r<>'A'               236
2510 IV=AV: GOSUB 780: GOSUB 3210: IF
 FK<>0 THEN RETURN                          164
2520 IF IV=6 THEN IF (AO=2) AND ((AN
AND &HFF00)=0) THEN B=&H36: GOSUB 630
: B=AN: GOTO 630 ELSE IF (AO=1) AND (
AV<>6) THEN B=&H70+AV: GOTO 630 ELSE
GOTO 3370 'ld (hl),..                       136
2530 REM ld a,..                              0
2540 IF AO<4 THEN OC=&H78: GOTO 1770        135
2550 ON AO-3 GOTO 3370,2560,2570,2580       168
2560 B=&H3A: GOSUB 630: B=AN: GOTO 68
0                                            27
2570 B=&HED: GOSUB 630: B=AV+&H10: GO
TO 630                                      241
2580 B=AV+8: GOTO 630                       135
2590 IN=AN: B=AV: GOSUB 630                 179
2600 GOSUB 780: GOSUB 3210: IF FK<>0
THEN RETURN                                  10
2610 IF (AO=2) AND ((AN AND &HFF00)=0
) THEN B=&H36: GOSUB 630: B=IN: GOSUB
 630: B=AN: GOTO 630 'ld (ix+d),n; ld
 (iy+d),n                                    78
2620 IF (AO=1) AND (AV<>6) THEN B=&H7
0+AV: GOSUB 630: B=IN: GOTO 630 ELSE
GOTO 3370 'ld (ix+d),r; ld (iy+d),r          31
2630 IV=AV: GOSUB 780: GOSUB 3210: IF
 FK<>0 THEN RETURN                          172
2640 ON AO GOTO 3370,2650,3370,2680,2
720,3370,3370                                60
2650 IF IV=4 THEN B=&HDD: GOSUB 630:
IV=2                                        207
2660 IF IV=5 THEN B=&HFD: GOSUB 630:
IV=2                                         12
2670 B=1+IV*16: GOSUB 630: B=AN: GOTO
 680                                        131
2680 IF (IV<>3) OR (AV<2) OR (AV=3) T
HEN GOTO 3370                               101
2690 IF AV=4 THEN B=&HDD: GOSUB 630          81
2700 IF AV=5 THEN B=&HFD: GOSUB 630         114
2710 B=&HF9: GOTO 630                        57
2720 IF (IV<2) OR (IV=3) THEN B=&HED:
 GOSUB 630: B=&H4B+16*IV: GOSUB 630:
B=AN: GOTO 680 'ld bc,(nn); ld de,(nn
); ld sp,(nn)                               159
2730 IF IV=4 THEN B=&HDD: GOSUB 630         137
2740 IF IV=5 THEN B=&HFD: GOSUB 630         198
2750 B=&H2A: GOSUB 630: B=AN: GOTO 68
0                                            18
2760 IN=AN: GOSUB 780                       214
2770 IF TK$<>"D" THEN B=&H32: GOSUB 6
30: B=IN: GOSUB 680: GOTO 2860 'ld (n
n),a                                         87
2780 IF AW=7 THEN GOTO 3370                 192
2790 IF (AW<2) OR (AW=3) THEN B=&HED:
 GOSUB 630: B=&H43+16*AW: GOSUB 630:
B=IN: GOSUB 680: GOTO 780 'ld (nn),bc
; ld (nn),de; ld (nn),sp                    131
2800 GOSUB 3440: B=&H22: GOSUB 630: B
=IN: GOSUB 680: GOTO 780                     82
2810 B=&HED: GOSUB 630                       93
2820 B=AV                                   221
2830 GOSUB 630: GOSUB 780                    57
2840 REM kijk of laatst gescande symb
ol 'A' was. Zo ja, scan volgend token         0
2850 REM en stop. Zo nee, stop met FK
=1.                                           0
2860 IF (TK$<>"H") OR (AW<>7) THEN GO
TO 3370                                     234
2870 GOTO 780                               142
2880 REM groep 14 (push,pop)                  0
2890 GOSUB 780: IF (TK$<>"D") OR (AW=
3) THEN GOTO 3370                            12
2900 IF (AW=4) OR (AW=5) THEN GOTO 29
20 'ix,iy                                    29
2910 B=OC+(AW AND 3)*16: GOSUB 630: G
OTO 780                                     172
```

```
2920 GOSUB 3440: B=OC+&H20: GOSUB 630
: GOTO 780                                 5
2930 REM groep 15 (ret)                    0
2940 GOSUB 3500: IF FK=0 THEN GOTO 78
0 ELSE FK=0: B=&HC9: GOTO 630            248
2950 REM groep 16 (rl,rlc,rr,rrc,sla,
sra,srl)                                   0
2960 GOSUB 780: GOSUB 3210: IF FK<>0
THEN RETURN                               34
2970 IF AO>3 THEN GOTO 3370               76
2980 ON AO GOTO 2990,3370,3000           218
2990 B=&HCB: GOSUB 630: B=OC+AV: GOTO
 630                                      47
3000 IX=OC+6: GOTO 3480                  252
3010 REM groep 17 (rst)                    0
3020 GOSUB 780: GOSUB 3160: IF NOT N
OR (AW<>AW AND &H38) THEN GOTO 3370      148
3030 B=OC+AW: GOSUB 630: GOTO 780        200
3040 REM groep 18 (djnz)                   0
3050 B=OC: GOSUB 630: GOSUB 780: GOTO
 3590                                    110
3060 REM groep 19 (out)                    0
3070 GOSUB 780: IF TK$<>"B" THEN GOTO
 3370                                    123
3080 GOSUB 780: GOSUB 3160: IN=N          34
3090 IF N THEN IF W THEN FK=2: RETURN
 ELSE B=&HD3: GOSUB 630: B=AW: GOSUB
630 ELSE IF (TK$<>"H") OR (AW<>1) THE
N GOTO 3370                              204
3100 GOSUB 780: IF TK$<>"E" THEN GOTO
 3370                                    182
3110 GOSUB 780: IF TK$<>"K" THEN FK=2
: RETURN                                  68
3120 GOSUB 780: IF (TK$<>"H") THEN GO
TO 3370                                  194
3130 IF IN THEN IF AW<>7 THEN GOTO 33
70 ELSE GOTO 780 'out (n),a               58
3140 B=&HED: GOSUB 630: B=&H41+AW*8:
GOSUB 630: GOTO 780                      183
3150 REM --getnum                          0
3160 IF TK$="L" THEN GOSUB 740 'labva
l                                         89
3170 N=(TK$="L") OR (TK$="C")            203
3180 W=N AND ((AW AND &HFF00)<>0)         88
3190 RETURN                              154
3200 REM --oper                            0
3210 FK=0                                211
3220 IF TK$="H" THEN IF AW>7 THEN AO=
6: IF AW=8 THEN AV=&H47: GOTO 780 ELS
E AV=&H4F: GOTO 780 ELSE AO=1: AV=AW:
 GOTO 780 'r                             173
3230 GOSUB 3160: IF N THEN AO=2: AN=A
W: GOTO 780 'n                           191
3240 IF TK$="D" THEN IF AW=7 THEN GOT
O 3370 ELSE AO=4: AV=AW: GOTO 780 'rr     87
3250 IF TK$<>"B" THEN GOTO 3370           94
3260 GOSUB 780: IF TK$="D" THEN IF AW
=2 THEN AO=1: AV=6: GOTO 3350 ELSE AO
=7: IF AW=0 THEN AV=2: GOTO 3350 ELSE
 IF AW=1 THEN AV=&H12: GOTO 3350 ELSE
 GOTO 3370 '(hl);(de);(bc)                 8
3270 GOSUB 3160: IF N THEN AO=5: AN=A
W: GOTO 3350 '(nn)                       235
3280 IF TK$<>"I" THEN GOTO 3370          208
3290 AO=3: AV=AW                         167
3300 GOSUB 780: GOSUB 3160: IF NOT N
THEN GOTO 3370                           218
3310 IF W THEN FK=2: RETURN              153
3320 AN=AW                               206
3330 REM check of volgende symbol ) i
s. Als oke, dan wordt het volgende         0
3340 REM symbol gelezen en gestopt, a
nders wordt met FK=1 gestopt.              0
3350 GOSUB 780: IF TK$="E" THEN GOTO
780                                      133
3360 REM stop met foutnr. 1                0
3370 FK=1: RETURN                        206
3380 REM emit av,ix,an. Wordt gebruik
t na aanroep van oper om (IX+d),           0
3390 REM (IY+d) af te handelen.            0
3400 B=AV: GOSUB 630: B=IX: GOSUB 630
: B=AN: GOTO 630                          83
3410 REM verwerk bc,de,hl of sp in oc
 en emit resultaat                         0
3420 B=OC+AW*16: GOSUB 630: GOTO 780     118
3430 REM als AW=4 wordt &HDD geemit,
als AW=5 wordt &HFD geemit                 0
3440 IF AW=4 THEN B=&HDD: GOTO 630       251
3450 IF AW=5 THEN B=&HFD: GOTO 630        56
3460 RETURN                              151
3470 REM zelfde als 13100, maar nu wo
rdt av,&HCB,an,ix geemit                   0
3480 B=AV: GOSUB 630: B=&HCB: GOSUB 6
30: B=AN: GOSUB 630: B=IX: GOTO 630      101
3490 REM --verw.kond.                      0
3500 FK=0: GOSUB 780                      92
3510 IF TK$="J" THEN IK=AW ELSE IF (T
K$="H") AND (AW=1) THEN IK=3 ELSE GOT
O 3370                                     3
3520 REM IK bevat nu kond.nr., met ko
nd.C = 3                                   0
3530 B=OC+IK*8: GOTO 630                 147
3540 REM --scan adres                      0
3550 FK=0: GOSUB 3160                      6
3560 IF NOT N THEN GOTO 3370              80
3570 B=AW: GOSUB 680: GOTO 780            69
3580 REM --scan rel.adres                  0
3590 FK=0: GOSUB 3160: IF NOT N THEN
GOTO 3370                                 21
3600 B=AW-CS-BP+BS-1: IF ((B>127) OR
(B<-128)) AND (PASS<>1) AND (FK=0) TH
EN FK=4: RETURN                           39
3610 IF B<0 THEN B=B+256                 164
3620 GOSUB 630: GOTO 780                 165
```

Lezers helpen lezers

In deze rubriek belanden die lezersvragen waar we zelf ook geen antwoord op weten, en de antwoorden die door andere lezers worden ingestuurd. Die vragen kunnen variëren van ongebruikelijke ML-routines, tot verkrijgbaarheid van materialen, boeken, en dergelijke.
Ook meningen van lezers, waar de redactie overigens niet verantwoordelijk voor is, kunnen in deze hoek terecht. Dus, als u wilt weten hoe u uw Wurlitzer jukebox met behulp van een MSX kunt besturen, om maar eens een dwarsstraat te noemen, waag er eens een briefje aan. Nee heb je, ja kun je krijgen.

LEZERSFORUM EN PROBLEMENHOEK

Het kan soms vreemd lopen in het leven van een LHL redactrice. 's middags heb je iemand aan de vragenlijn, die problemen heeft met het grafisch aansturen van de Star LC-10 printer, 's avonds scheur je een envelop open...

DESPAT

En daarin bevinden zich, naast fraaie fractals, de nodige oplossingen. Ons toegestuurd door Johan Meuwissen uit Limbricht.
DESPAT is een waardige opvolger van SEPPAT. DESPAT past Desiplus aan voor gebruik in combinatie met de Star LC-10. Het werkt wel ietsje anders dan SEPPAT, want DESPAT moet iedere keer vóór Desiplus gerund worden. Er wordt namelijk een hook omgebogen, naar ik aanneem om onderweg de tekens die naar de printer gestuurd worden te kunnen afvangen.
Ook nu wordt van harte aanbevolen om met een **kopie** van Desiplus te werken. Er wordt namelijk ook direct in het programma op de schijf ingegrepen. Voor het overige wijst de weg zichzelf. Kleurenprinten wordt – nog? – niet ondersteund.

Johan stuurde ook een oplossing voor Dynamic Publisher met de LC10 mee, maar gezien de lengte van de listing bewaar ik die even tot het eind.

WBASS-2-2

Uit de brief van C. Zijlstra wordt niet helemaal duidelijk wàt nu precies wel en niet werkt bij het laten 'verdwijnen' van WBASS2. Waarschijnlijk werken de genoemde PAGE-instructies wel, maar kun je er niet zoveel mee aanvangen.
Je kunt namelijk niet zomaar straffeloos de pages nul en drie uitschakelen! Hier zitten respectievelijk de ROM en de systeempagina. Wil je iets laten verdwijnen, dan kun je je dus het beste beperken tot pagina één en twee.

Normaal gesproken werk je voornamelijk op de pagina's twee en drie. Dus het zou – kort samengevat – handig kunnen zijn om pagina twee naar de memory mapper te verwijzen met behulp van één van de beschreven PAGE-opdrachten.
Echt handig vind ik dat zelf niet, WBASS werkt zo snel, dat je makkelijk eerst naar disk kunt assembleren, om daarna het resultaat te bekijken.
Nog even wat tips:
In de AUTOEXEC.BAS kun je wat handige functietoetsen klaarzetten, bijvoorbeeld:

```
KEY 1, CHR$(3) + CHR$(12) + CHR(13)
```

Dit levert ^C en scherm schoon.

Het volgende had eigenlijk in de handleiding moeten staan.
Je kunt met SET diverse zaken instellen. Bijvoorbeeld:

```
SET/H,V,"#".
```

Het is echter wel wat lastig om dat elke keer te moeten doen. De oplossing hiervoor is: geef het programma de gewenste instelling en schrijf het dan met BSAVE weg – alvorens PAGE te gebruiken! – met:

```
BSAVE "WBASS2.BIN",#4000,#75BF
```

Deze twee adressen kun je van te voren nog even controleren met het commando ADDR.
Met vriendelijke groet,

Jan van der Meer, Groningen

Hopelijk spreekt deze brief voor zich, voor de gebruikers van WBASS-2. Zoals reeds gemeld heb ik, waar het over assembler gaat wel de klok horen luiden – oh nee, dat komt straks. Dank aan Jan van der Meer. En als er nog vragen zijn: schrijf!

Datum draaien

Al weer enige tijd geleden benaderde Simon C. Dekker – wel bekend van diverse muzikale inzendingen – ons met twee probleempjes, die nu toch maar eens opgelost moeten worden.

Het eerste betreft het formaat waarin de datum door een MSX computer wordt onthouden. Op 31 december 1999 is het makkelijk af te lezen waar de dag- maand- en jaar aanduiding – afgekort DD/MM/JJ – staan, maar op 1 januari

```
10 ' DESPAT                                                                       0
20 ' past Desiplus aan voor de Star LC-10 printer                                 0
30 ' MSX Computer Magazine                                                        0
40 '                                                                              0
50 ' ingezonden door Johan Meuwissen, Limbricht                                   0
60 '                                                                              0
70 SCREEN 0: WIDTH 80: KEY OFF: COLOR 15,4,4                                     69
80 PRINT "Dit Programma past Designer Plus aan voor het afdrukken van schermen o
p de     STAR LC-10 printer."                                                   233
90 PRINT "Zorg ervoor dat een disk met een KOPIE (!!!!) van Designer Plus, met d
e file-   naam DESIPLUS.COM, in de drive is geplaatst, waarop ook DOS staat."    87
100 PRINT "Dit programma ALTIJD eerst opstarten voordat met Designer Plus geprin
t wordt. Ermoet namelijk een hook worden omgebogen"                             207
110 ON ERROR GOTO 220: OPEN "desiplus.com" FOR INPUT AS#1: CLOSE: ON ERROR GOTO
0                                                                               157
120 OPEN "DESIPLUS.COM" AS #1 LEN=1: FIELD #1,1 AS I$: I=14546: RESTORE 230     249
130 GET 1,I: IF ASC(I$)<>176 THEN B$=CHR$(127) ELSE B$="was al": GOTO 150        40
140 READ A$: IF A$<>"*" THEN A=VAL(A$): LSET I$=CHR$(A): PUT #1,I: I=I+1: GOTO 1
40                                                                                3
150 CLOSE #1: PRINT: PRINT: PRINT "Designer plus "B$" aangepast."               109
160 PRINT "Hook &HFFB6 wordt omgebogen....": RESTORE 240                          2
170 READ A$: IF A$<>"*" THEN POKE &HFA75+T,VAL("&H"+A$): T=T+1: GOTO 170        171
180 POKE &HFFB6,&HC3: POKE &HFFB7,&H75: POKE &HFFB8,&HFA                        153
190 PRINT: PRINT "Designer Plus wordt geladen": A$="DESIPLUS"+CHR$(13)          158
200 POKE &HF3FA,&HF0: POKE &HF3FB,&HFB: POKE &HF3F8,&HF9: POKE &HF3F9,&HFB       26
210 FOR T=1 TO 9: POKE &HFBEF+T,ASC(MID$(A$,T,1)): NEXT T: POKE &HF346,1: CALL S
YSTEM                                                                            15
220 PRINT "DESIPLUS.COM is niet aanwezig op deze disk, plaats disk met DESIPLUS.
COM en drukop F5.": END                                                          39
230 DATA 176,216,204,24,00,000,00,000,00,00,00,00,00,000,254,216,90,1,224,*     221
240 DATA ED,43,00,FE,06,08,CB,2F,CB,11,05,C2,7B,FA,79,ED,4B,00,FE,C9,*           21
```

2001 lezen we waarschijnlijk: 01/01/01 en dan is het dus volslagen onduidelijk, waar DD, MM of JJ zich bevinden.

De heer Dekker zond een listinkje uit de handleiding mee, dat zou moeten helpen om hier wijs uit te worden, maar – zoals hij terecht opmerkte – dat werkte niet goed. Wij hebben het even iets verder uitgezocht en dat heeft geleid tot de listing DRDAT. DRDAT zoekt op hoe-om de datum in het geheugen staat en drukt vervolgens de juist geordende datum af. Althans in de ogen van de meeste Nederlanders.
In bit 5 en 4 op adres &H002B is vastgelegd in welk formaat de datum wordt bewaard.

```
10 ' DRDAT                                  0
20 '                                        0
30 ' MSX Computer Magazine                  0
40 ' voor Simon C. Dekker                   0
50 '                                        0
60 B=PEEK(&H2B)                           220
70 A$="00000"+BIN$(B)                      32
80 C$=RIGHT$(A$,6)                         49
90 D$=LEFT$(C$,2)                          93
100 GET DATE E$                            54
110 IF D$="00"THEN E$=MID$(E$,7,2)+"/
"+MID$(E$,4,2)+"/"+MID$(E$,1,2)           155
120 IF D$="01"THEN E$=MID$(E$,4,2)+"/
"+MID$(E$,1,2)+"/"+MID$(E$,7,2)            67
130 PRINT "de datum: " E$                 174
```

De mogelijke waarden zijn:

00 : JJ/MM/DD
01 : MM/DD/JJ
10 : DD/MM/JJ

Aangezien dit zich in het ROM bevindt is het niet mogelijk een welgemikte POKE uit te voeren om het formaat te wijzigen. Maar we kunnen het wel uitlezen en op grond van de gevonden bit instelling de datumstring aanpassen.

Aldus doet DRDAT. Het gedoe met BIN$ en de nullen is bedoeld om zeker te zijn dat we de onderste zes bits op een rijtje hebben; zouden de hoogste bits op nul staan, dan is BIN$ evenredig korter, hetgeen tot illegal function call's zou kunnen leiden.
Daarna is een korte goocheltruc met MID$ genoeg om de dagaanduiding naar voren te dirigeren.

Toch wil ik weer meer: het is natuurlijk veel mooier als je met behulp van een hook of ander machinetaal-gereed-

```
10 ' KRKKLK                                   0
20 '                                          0
30 ' MSX Computer Magazine                    0
40 ' Een probleem van Simon C. Dekker         0
50 '                                          0
60 CLS: SCREEN 0: KEY OFF: COLOR 10,1
,1: WIDTH 37                                 95
70 LOCATE 12,6                              251
80 PRINT "- KERKKLOK -"                     216
90 A$="T100S9M6000002G1E1G1E1"               29
100 PLAY A$: PLAY A$                        198
110 B$="T10005L2S10M30000ECECECEC"          179
120 PLAY B$: PLAY B$                        231
130 PLAY A$,B$: PLAY A$,B$                  167
```

schap zorgt dat altijd als de datum wordt opgevraagd deze onmiddellijk in de gewenste vorm gegoten wordt.
Wie o wie?

KRKKLK

En dit zijn dan de beloofde luiklokken. De listing KRKKLK is afkomstig van de heer Dekker. Hier doet zich een vreemd verschijnsel voor, als beide 'klokken' tegelijk geluid worden.
De eerste klok geeft een zeer langzame slag, te beluisteren bij PLAY A$ in regel 100. De tweede slaat, zoals het hoort, hoger en sneller, luister maar naar B$ in regel 120. Worden nu deze twee strings op twee kanalen naast elkaar gezet —

```
10 REM BASIC-LOADER                                                          0
20 REM                                                                       0
30 REM Dit programma is grotendeels gegenereerd door datmak                  0
40 REM                                                                       0
50 REM Het bevat de DATA-weergave van lc10*.sys bestanden                    0
60 REM                                                                       0
70 RESTORE 1000: GOSUB 130: RESTORE 1000: GOSUB 230                         45
80 RESTORE 2000: GOSUB 130: RESTORE 2000: GOSUB 230                         95
90 RESTORE 3000: GOSUB 130: RESTORE 3000: GOSUB 230                        145
100 RESTORE 4000: GOSUB 130: RESTORE 4000: GOSUB 230                       241
101 RESTORE 5000: GOSUB 130: RESTORE 5000: GOSUB 230                        39
102 RESTORE 6000: GOSUB 130: RESTORE 6000: GOSUB 230                        93
110 RESTORE 7000: GOSUB 130: RESTORE 7000: GOSUB 230                       140
120 CLS:PRINT "klaar!": END                                                 23
130 READ F1$,RL,FL: N=0: CK=0: NC=0: VL=0                                  141
140 CLS: WIDTH 37: PRINT "Deze Basic-lader maakt het bestand ofprogramma ";F1$;"
 aan."                                                                      89
150 PRINT: PRINT "Dataregels worden eerst gecontroleerd": PRINT "Even geduld aub
...."                                                                      143
160 ' check data-regels *************                                        0
170 READ A$: N=N+1: NC=NC+1: IF VL=1 THEN NC=NC+VAL("&h"+A$)-3: VL=0 ELSE IF A$=
"**" THEN VL=1                                                             232
180 CK=CK+ASC(LEFT$(A$,1))+ASC(RIGHT$(A$,1))*2                             233
190 IF NMODRL=0 THEN READ CR$: IF CK=VAL("&H"+CR$) THEN CK=0 ELSE GOTO 380 238
200 IF NC<FL THEN GOTO 170                                                  76
210 READ CR$: IF CK=VAL("&H"+CR$) THEN CK=0 ELSE GOTO 380                  120
220 RETURN                                                                 192
230 ' maak bestand *****************                                         0
240 OPEN F1$ AS #1 LEN=1                                                   159
250 FIELD #1,1 AS I$                                                        76
260 RESTORE                                                                189
270 PRINT: PRINT "Aan het werk..."                                         211
280 READ F1$,RL,FL: N=0: NC=0                                              205
290 READ A$: N=N+1: NC=NC+1: IF NMODRL=0 THEN READ CR$                      99
300 IF A$<>"**" THEN LSET I$=CHR$(VAL("&H"+A$)): PUT #1: GOTO 360            9
310 READ A$: N=N+1: BT=VAL("&H"+A$): IF NMODRL=0 THEN READ CR$             171
320 READ A$: N=N+1: BV=VAL("&H"+A$): IF NMODRL=0 THEN READ CR$             217
330 FOR N1=1 TO BT                                                         213
```

```
340    LSET I$=CHR$(BV): PUT #1                                                201
350 NEXT N1: NC=NC+BT-1                                                        223
360 IF NC<FL THEN GOTO 290                                                     163
370 CLOSE: PRINT: PRINT "Klaar met  ";F$: RETURN                               243
380 PRINT "Fout gevonden in regel:"                                            200
390 I=PEEK(-2360)+256*PEEK(-2359)-1: FOR F=I TO 0 STEP-1: IF PEEK(F)<>0 THEN NEX
T F ELSE PRINT PEEK(F+3)+256*PEEK(F+4)                                           8
400 STOP                                                                       228
1000 DATA lc10-60.sys, 20 , 226                                                197
1010 DATA **,03,07,00,00,07,**,03,00,07,00,00,E0,01,00,FF,01,FF,20,0E,C1B      188
1020 DATA 32,37,2C,36,34,2C,32,37,2C,35,31,2C,32,34,**,12,00,20,0D,32,C72      164
1030 DATA 37,2C,34,32,2C,30,2C,32,32,34,2C,31,**,13,00,20,0A,32,37,2C,C7D       97
1040 DATA 37,34,2C,32,2C,31,33,**,16,00,20,0B,32,37,2C,31,30,36,2C,34,C6C      185
1050 DATA 2C,31,30,**,15,00,20,00,32,37,2C,31,32,**,1B,00,20,01,31,**,BC1      129
1060 DATA 1F,00,03,03,00,20,39B                                                 46
2000 DATA lc10-72.sys, 20 , 226                                                248
2010 DATA **,03,07,00,00,07,**,03,00,07,00,00,40,02,00,FF,01,FF,20,0E,C0C      210
2020 DATA 32,37,2C,36,34,2C,32,37,2C,35,31,2C,32,34,**,12,00,20,0C,32,C70      209
2030 DATA 37,2C,34,32,2C,35,2C,36,34,2C,32,32,**,13,00,20,0A,32,37,2C,C91      137
2040 DATA 37,34,2C,32,2C,31,33,**,16,00,20,0B,32,37,2C,31,30,36,2C,34,C6C      186
2050 DATA 2C,31,30,**,15,00,20,00,32,37,2C,31,32,**,1B,00,20,01,31,**,BC1      130
2060 DATA 1F,00,03,03,00,20,39B                                                 47
3000 DATA lc10-80.sys, 20 , 226                                                231
3010 DATA **,03,07,00,00,07,**,03,00,07,00,00,80,02,00,FF,01,FF,20,0E,C10      108
3020 DATA 32,37,2C,36,34,2C,32,37,2C,35,31,2C,32,34,**,12,00,20,0D,32,C72      166
3030 DATA 37,2C,34,32,2C,34,2C,31,32,38,2C,32,**,13,00,20,0A,32,37,2C,C8D      191
3040 DATA 37,34,2C,32,2C,31,33,**,16,00,20,0B,32,37,2C,31,30,36,2C,34,C6C      187
3050 DATA 2C,31,30,**,15,00,20,00,32,37,2C,31,32,**,1B,00,20,01,31,**,BC1      131
3060 DATA 1F,00,03,03,00,20,39B                                                 48
4000 DATA lc10-90.sys, 20 , 226                                                248
4010 DATA **,03,07,00,00,07,**,03,00,07,00,00,D0,02,00,FF,01,FF,20,0E,C1C       12
4020 DATA 32,37,2C,36,34,2C,32,37,2C,35,31,2C,32,34,**,12,00,20,0D,32,C72      167
4030 DATA 37,2C,34,32,2C,36,2C,32,30,38,2C,32,**,13,00,20,0A,32,37,2C,C8F       97
4040 DATA 37,34,2C,32,2C,31,33,**,16,00,20,0B,32,37,2C,31,30,36,2C,34,C6C      188
4050 DATA 2C,31,30,**,15,00,20,00,32,37,2C,31,32,**,1B,00,20,01,31,**,BC1      132
4060 DATA 1F,00,03,03,00,20,39B                                                 49
5000 DATA lc10-120.sys, 20 , 226                                                60
5010 DATA **,03,07,00,00,07,**,03,00,07,00,00,C0,03,00,FF,01,FF,20,0E,C1D       90
5020 DATA 32,37,2C,36,34,2C,32,37,2C,35,31,2C,32,34,**,12,00,20,0D,32,C72      168
5030 DATA 37,2C,34,32,2C,31,2C,31,39,32,2C,33,**,13,00,20,0A,32,37,2C,C8B       29
5040 DATA 37,34,2C,32,2C,31,33,**,16,00,20,0B,32,37,2C,31,30,36,2C,34,C6C      189
5050 DATA 2C,31,30,**,15,00,20,00,32,37,2C,31,32,**,1B,00,20,01,31,**,BC1      133
5060 DATA 1F,00,03,03,00,20,39B                                                 50
6000 DATA lc10s120.sys, 20 , 226                                                87
6010 DATA **,03,07,00,00,07,**,03,00,07,00,00,C0,03,00,FF,01,FF,20,0E,C1D       91
6020 DATA 32,37,2C,36,34,2C,32,37,2C,35,31,2C,32,34,**,12,00,20,0F,32,C76       81
6030 DATA 37,2C,34,32,2C,32,2C,31,2C,31,39,32,2C,33,**,11,00,20,0A,32,C7D      204
6040 DATA 37,2C,37,34,2C,32,2C,31,33,**,16,00,20,0B,32,37,2C,31,30,36,C72      225
6050 DATA 2C,34,2C,31,30,**,15,00,20,00,32,37,2C,31,32,**,1B,00,20,01,C01      224
6060 DATA 31,**,1F,00,03,03,00,20,4AE                                          158
7000 DATA lc10-240.sys, 20 , 226                                               112
7010 DATA **,03,07,00,00,07,**,03,00,07,00,00,80,07,00,FF,01,FF,20,0E,C1A       72
7020 DATA 32,37,2C,36,34,2C,32,37,2C,35,31,2C,32,34,**,12,00,20,0D,32,C72      170
7030 DATA 37,2C,34,32,2C,33,2C,31,32,38,2C,37,**,13,00,20,0A,32,37,2C,C95      138
7040 DATA 37,34,2C,32,2C,31,33,**,16,00,20,0B,32,37,2C,31,30,36,2C,34,C6C      191
7050 DATA 2C,31,30,**,15,00,20,00,32,37,2C,31,32,**,1B,00,20,01,31,**,BC1      135
7060 DATA 1F,00,03,03,00,20,39B                                                 52
```

regel 130 – dan hoor je duidelijk dat de grote klok telkens gelijk met de kleine klok een keer extra wordt aan geslagen. Is dit nu gehoor-bedrog? Of wat doen we hieraan?

Dynamic Publisher

Als tweede zeer interessante printerhulp zond Johan Meuwissen zeven piepkleine .SYS bestanden op schijf. Deze zijn bedoeld om te gebruiken als printerdrivers voor de Star LC-10 in Dynamic Publisher. In Dynamic Publisher kun je onder de optie 'systeem' naar 'instelling' gaan, om aldaar te besluiten een .SYS bestand te 'laden'. Vervolgens kun je, om een zo mooi mogelijk resultaat te krijgen, bij 'opties' kiezen voor 'uitprinten' en daar kiezen voor 'dubbel aanslaan'.

De zeven .SYS bestandjes onderscheiden zich van elkaar voornamelijk door de horizontale dichtheid bij het uitprinten. Dit betekent dat de getekende figuur horizontaal wordt uitgerekt of in elkaar gedrukt. Het mooiste effect bereik je naar onze smaak met LC10-120.SYS. Maar het is altijd prettig om te kunnen kiezen en experimenteren.
Om de bestandjes als listing te kunnen publiceren hebben we ons vertrouwde DATMAK maar weer eens uit de kast gehaald. Natuurlijk zou het onzinnig zijn om voor elk bestandje weer de eerste veertig regels in te typen. Daarom heb ik de verkregen dataregels achter elkaar gehangen en de regels 70 tot en met 110 ingevoegd.
Wie zeven bestanden teveel vindt kan zelf een paar van deze regels overslaan en de corresponderende dataregels weglaten. De oplettende listingtypist(e) zal opmerken dat sommige dataregels exact herhaald worden. Dat klopt slechts gedeeltelijk, vaak is er net één getalletje toch anders.
Ik ben aan de veilige kant gebleven door botweg het volledige DATMAK resultaat neer te zetten. Waarschijnlijk kan een hoop typewerk bespaard worden door regels te kopiëren naar een hoger duizendtal. Het is dan echter wel zaak om met behulp van ICP en het controlegetal aan het eind van iedere regel de nodige correcties aan te brengen!

Concluderend durf ik te stellen dat de oproep tot printeroplossingen langzamerhand tot prima resultaten leidt. En dan heb ik het nog niet eens gehad over APEP. APEP is een uiterst pretentieus programma: de eerste drie letters staan voor Anti Printer Ellende. Het werd ons toegezonden door Vick van Acht uit Sint Oedenrode.

De ellende met dergelijke uitgebreide pakketten is dat er tijd en een hoop werk nodig is om het grondig te testen. Kortom, in de Paasvakantie zullen de PD-redacteur en ik geen eieren zoeken. Want APEP is te groot voor deze rubriek: houdt het PD aanbod de komende tijd in de gaten.
Inmiddels wordt uit de reacties op reeds gepubliceerde patches duidelijk dat de behoefte aan méér groot is. Blijf insturen, zo bid ik u!

Klantsys demodisk

Bent u na MSX Computer Magazine 37, Software Gids 1 en MSX Club Magazine 28 nog niet overtuigd van de mogelijkheden van Klantsys?

Bestelt u dan tegen een ware wegwerpprijs een demo van Klantsys. Daarmee bent u in staat alle funkties van het pakket te gebruiken, met die beperking dat u slechts een kleine administratie kunt aanmaken op uw diskette.

Hoewel Klantsys eenvoudig in het gebruik is, wordt bij de demo de komplete gebruiksaanwijzing meegeleverd.

U kunt alle schermen bekijken om uiteindelijk tot de ontdekking te komen dat Klantsys zeer gebruikersvriendelijk is.

Zelfs de printer wordt in de demo volledig ondersteund, zodat u uw 'testadministratie' ook op papier tot uitdrukking kunt laten komen!

Het demopakket kost slechts F10,00 (inclusief BTW en verzendkosten). Dit bedrag wordt in mindering gebracht op de prijs van Klantsys als u tot aanschaf van het normale pakket overgaat.

Brainchild 03486-4419 (tussen 19.00-20.00u)

(Brainchild - Twijnen 48 - 3421 JP Oudewater)

MSX SUCOM MSX

KARTUIZERSVEST 109
2500 LIER BELGIE
TEL: 03/489.28.81

Steeds in voorraad:
De allernieuwste MSX2/MSX2+ games.
Een greep uit ons gamma:
- Space mambow (Konami) SCC,rom
- Aleste 2 (Compile) FM-PAC,3 disc's
- Hydefos (Hertz) FM-PAC,2 disc's
- Gunship (Micro prose) disc
- Pacmania (Namcot) FM-PAC, disc
- Xevious (Namcot) rom
- R-type (Irem) rom
Wordt verwacht:
- Family boxing
- Double dragon
- Batman the movie
- Rune worth ,enz.
Professioneel,schitterend,onmisbaar :
- MSX-DOS 2.20 compleet(HSH) rom+disc
- MSX-DOS 2 TOOLS (Ascii) disc
- MSX C-compiler (Ascii) disc
- MSX C-library (Ascii) disc
- Dempa sensor kit (Dempa micom soft)
Hardware:
MSX2/+ computers, muizen, scanners, printers, track balls,joysticks/cards

Panasonic FS PC1, 48 dots kleurenprinter



Enigszins verbaasd waren we wel, toen er een 48 dots printer op de redactie werd binnen gedragen met de aanduiding MSX op de verpakking. Een enkeling begon ronduit te kwijlen, toen er op de verpakking ook nog eens het woordje 'COLOR' bleek te staan. Tussen allerlei onbegrijpelijke Japanse kreten was dat dan ook precies alles wat we konden ontcijferen, behalve het merk en typenummer, de Panasonic FS PC1.

JAPANSE MSX KLEURENPRINTER GETEST

Een test-apparaat, met andere woorden. Japanse import van het MSX-Centrum in Amsterdam. Een printer die niet in de reguliere handel verkrijgbaar is, maar op bestelling geleverd kan worden. Bovendien, een machine die niet voor 220 Volt gebouwd is, zodat er een extra transformator nodig is om het ding aan de praat te krijgen.

Onleesbaar

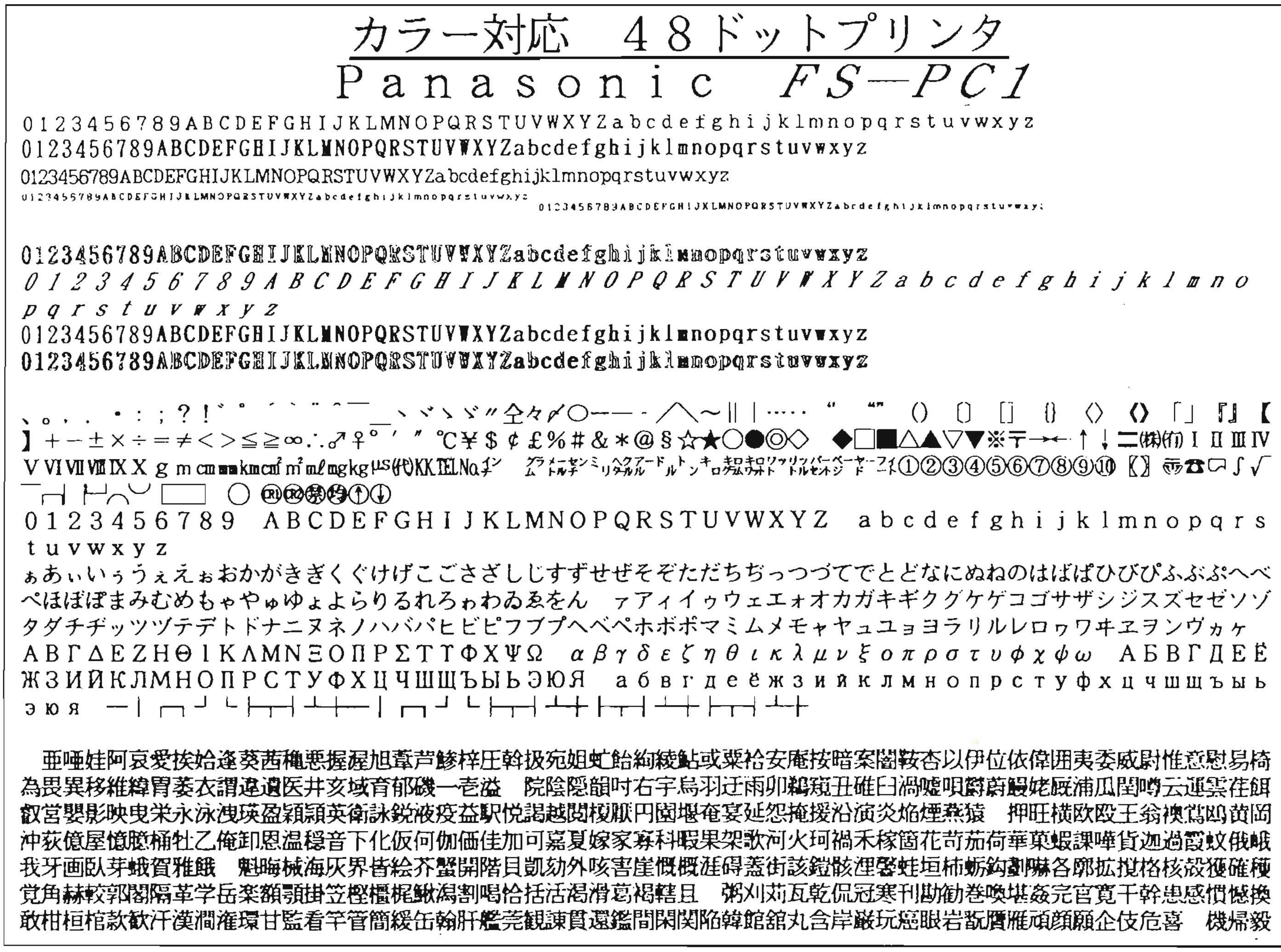
Na het apparaat eens goed van alle kanten te hebben bekeken kwam de volgende stap in onze test: het bestuderen van de handboeken. Een goed verzorgd boekwerkje, maar helemaal in het Japans. En daar kunnen we nog steeds niet goed aan wennen, al wordt er momenteel met een schriftelijk opleidingsinstituut onderhandeld over een cursus Japans voor MSX-redacteuren.

Gelukkig is het een echte MSX-printer, wat wil zeggen dat de ons bekende stuurcodes voor normale MSX-printers naar behoren zullen werken. De extra's die deze printer heeft, zoals bijvoorbeeld een complete JIS-1 en JIS-2 karakterset zullen wat moeilijker te bereiken zijn, bij gebrek aan leesbare documentatie. Maar de codes die noodzakelijk zijn om deze karakters op papier te krijgen zijn toch wel in het boekje terug te vinden, aangezien men tussen alle Japanse tekst zo hier en daar opeens wat voor ons leesbare kreten opneemt. Het is echter wel puzzelen en uitproberen.

Thermisch

Ook blijkt uit die handleiding dat het hier gaat om een 'thermal transfer' printer, geen gewone dot-matrixprinter. De printkop van deze printer bevat geen mechanische delen, zoals printnaalden, maar een groot aantal – 48 om precies te zijn – 'hot-spots'. Dat zijn elementjes op die printkop die electrisch verhit kunnen worden, waardoor het warmtegevoelige papier op die plek zwart wordt.

Het resultaat is dat deze printer bijna zonder geluid te maken op thermisch papier kan printen. Het nadeel van ther-



Afbeelding 1a: Zelftest (is tot 93% verkleind)

mische printers is echter dat het afdrukken niet onbeperkt houdbaar zijn, ze verbleken na een tijdje. Bovendien is thermisch papier wat kostbaar en vrij dun vergeleken met gewoon printerpapier.

Maar deze Panasonic is niet een simpele thermische printer, het is zoals reeds gezegd een 'thermal transfer' printer. En dat 'transfer' biedt heel veel mogelijkheden. Er zijn namelijk printerlinten die niet zoals gebruikelijk op mechanische druk reageren, zoals bij de gewone dot-matrix, maar die op hitte reageren. Er kan een eenmalig carbonlint in de FS PC1 worden geplaatst, zodat er ook op gewoon papier gedrukt kan worden. Een wat dure oefening, vergeleken met het ook al niet goedkope thermische papier, maar het levert wel een prachtige afdruk op, op gewoon papier.

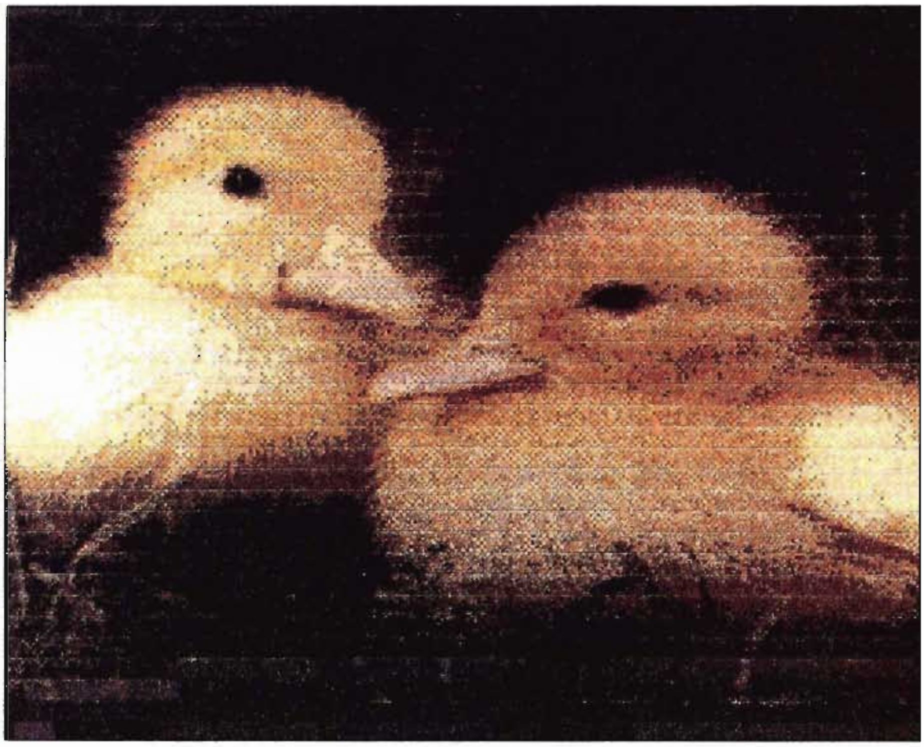
Overigens, kettingpapier past er niet in. De printer is bestemd voor losse vellen, terwijl men eventueel ook rollen thermisch papier — zoals die voor fax-apparaten heel gewoon zijn — kan gebruiken.

Kwaliteit

De letters die met de meegeleverde zwarte lintcassette op papier komen zijn van uitzonderlijke kwaliteit, erg scherp en bijzonder mooi. Alleen, erg snel is het niet. Zelfs de standaard-letter verschijnt slechts met een snelheid van zo'n 40 tot 50 tekens per seconde op papier.

Afbeelding 1b: Zelftest in kleurenmode

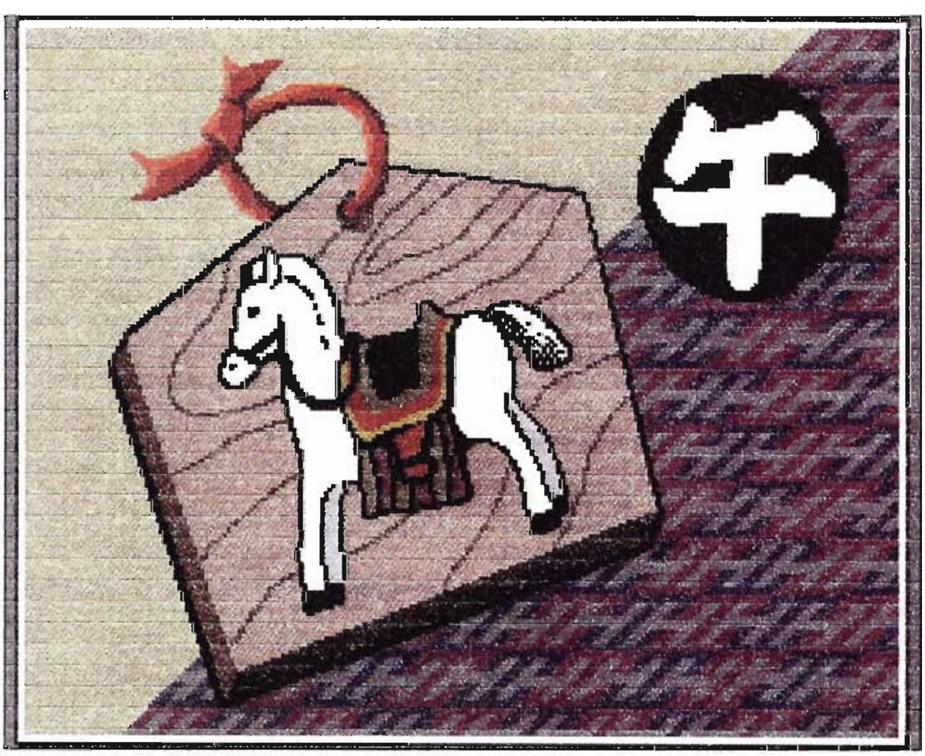




Figuur 2a: Een gedigitaliseerd plaatje

Dat standaard-lettertje is overigens ook al een goed gevormde letter, waarbij de afzonderlijke puntjes niet meer te onderscheiden zijn. De pitch van deze letter staat op 10, hetgeen wil zeggen dat er 10 letters per inch afgedrukt worden. Een prima letter voor bijvoorbeeld listings. Daarnaast kent de printer nog veel meer schriftsoorten. Niet precies wat we hier gewend zijn, want Pica en Elite hebben we niet kunnen ontdekken. Maar wel weer bijvoorbeeld een heel aardig ogende outline-letter, een soort hol letterbeeld. Ook heeft men een behoorlijk compleet symbolenfont ter beschikking. Als voorbeeld is bij dit artikel een stukje van de zelftest afgedrukt van deze printer, in afbeelding 1. Als bij het aanzetten van de printer de twee rechter toetsen op het frontpaneel ingedrukt worden zal de FS-PC1 alle mogelijkheden laten zien.

Figuur 2b: Voorbeeld plaatje



JIS

En dat zijn er heel wat! Het uitgebreide ROM van de printer herbergt niet alleen de standaard MSX-tekenset, maar ook de van de MSX2+ bekende JIS-tekens. Dat zijn meerdere Japanse en Chinese tekensets, aangevuld met de al genoemde symbolen en — om maar wat te noemen — Russische en Griekse alfabetten. De 48 dots staan garant voor een kwalitatief erg goede letter!
Alleen, het praktisch nut van al die karakters zal hier in Europa vrij beperkt zijn. Leuk om te hebben, maar tenzij men Japans studeert lijkt het ons niet erg bruikbaar. Ook de op zich wel zinnige zaken, zoals het Griekse of Russische alfabet zijn voor zover wij hebben kunnen ontdekken niet zonder meer vanuit de hier bekende MSX tekstverwerkers te gebruiken. Maar wie weet wat er nog te verwachten is uit Japan.

Kleur

Het mooiste hebben we voor het laatst bewaard. De FS-PC1 is namelijk een kleurenprinter.De tweede lintcassette die in de verpakking zit is er een met gekleurd carbonlint. De basiskleuren op dit lint zijn zwart, paars, rood, magenta, groen, blauw en geel. Deze kleuren zitten overigens niet boven elkaar zoals bij een aantal bekende kleurenprinters, maar achter elkaar. Door menging van deze basiskleuren zijn een groot aantal mengkleuren beschikbaar.
Om in kleur te kunnen printen moet er op het front van de printer een schakelaar op kleur worden gezet. Ook in de kleurenmode kan de testfunctie worden opgeroepen, door bij het aanzetten de twee rechtertoetsen op de printer ingedrukt te houden. Er wordt dan een overzicht van de beschikbare basiskleuren op papier gezet.

Screens printen

Bij de printer wordt ook nog een diskette meegeleverd waarop zich een programma bevindt waarmee plaatjes — waarschijnlijk alleen screen 8 — in kleur zijn af te drukken. Dit programma werkt

echter alleen op een MSX2+ computer, voor een MSX2 zal men een eigen screendump moeten schrijven. Op de disk staan een aantal schermbeelden die als voorbeeld dienen, maar ook andere screen 8 beelden konden we zonder problemen afdrukken.

De resultaten van de voorbeelden zijn verbluffend: mooie kleuren, hoge resolutie. Alleen de snelheid valt wat tegen. Voor plaatjes die bestaan uit een hoop verschillend gekleurde vlakken moet je toch al gauw rekenen op 30 tot 50 minuten printtijd.
Als tijdens het printen het einde van de cassette wordt bereikt wordt – in het Japans natuurlijk – gevraagd om de lintcassette om te draaien. Na een druk op RETURN of op de linker muisknop wordt het printen dan weer hervat.

Onze eigen experimenten toonden aan dat men die voorbeelden, die met de printer meegeleverd werden, wel had uitgezocht op hun afdrukbaarheid. De gedigitaliseerde pulletjes – zie figuur twee – komen toch net iets minder uit de verf.

Ook viel het ons op dat het lint – althans het kleurenlint – erg snel opraakt. Zo hebben we een screendump gemaakt van een fraaie adelaar, waar als een soort negatief een eerder gemaakte dump in viel te herkennen. Blijkbaar was de kleurstof na een keer al zover opgebruikt dat bij die tweede dump er een soort spookbeeld verscheen. De kwaliteit is goed, maar de men zal voor het beste resultaat liefst steeds een nieuw lint moeten nemen. En dat zal toch een beetje in de papieren gaan lopen.

Voltage

Met de groeiende stroom Japanse importen op hardwaregebied willen we toch even waarschuwen voor de risico's, die hier aan verbonden zijn.

In Japan hanteert men nu eenmaal een ander voltage dan we hier in Nederland hebben, hetgeen er toe leidt dat in veel gevallen allerlei adapters gebruikt zullen worden. Zo'n adapter is in feite een transformator, die het Nederlandse voltage – nog steeds 220 Volt, hoewel men dat gaat opvoeren tot 230 Volt in de loop van de komende paar jaar – omtovert tot de 100 Volt die in Japan standaard zijn.En daarbij kan er heel wat mis gaan. Zo is een dergelijke adapter niet echt makkelijk verkrijgbaar. Losse transformatoren zijn eenvoudiger te vinden, reden waarom men maar al te vaak – ook op de redactie – losse trafo's met simpele kroonsteen-aansluitingen ziet. Een gevaarlijke oplossing, zo'n trafo **moet** in een kastje zitten.
Bovendien, de omzetting van 220 naar 100 Volt is ook niet echt gemeengoed. Van 220 naar 110 Volt is wel redelijk makkelijk te vinden, maar dan krijgt de Japanse apparatuur wel een voortdurende te hoge voedingsspanning. Vaak staat in de technische specificatie '100 Volt, +/- 10%'. Die 110 Volt is dus altijd al tien procent meer dan waar het apparaat voor gebouwd is. En ook het Nederlandse lichtnet wil wel eens fluctueren qua spanning, waarbij 230 Volt helemaal geen uitzondering is.
En als we ons dan ook nog even bedenken dat die officiële 220 Volt in de loop van de komende jaren langzaam maar zeker zal worden opgevoerd tot 230 Volt, in het kader van de Europese eenwording...

Als we het voltageprobleem al onder de knie hebben komt er nog een potentiële bron van ellende om de hoek kijken: de capaciteit van de omzetter. Als de trafo bloedheet wordt, dan is dat een slecht teken. En veel van de omzettertjes die worden aangeboden zijn eigenlijk alleen bedoeld voor een scheerapparaat, die kan je na een paar uur gebruik niet meer aanraken van de hitte! Oftewel, de trafo dient ruim bemeten te zijn.

Als de technische documentatie van een Japans apparaat meldt dat het een 50 Watt opneemt, dan dient de trafo minimaal 75 Watt aan te kunnen. Transformatoren zijn namelijk niet voor de volle honderd procent efficient, er treden altijd verliezen op. Meestal zijn dat verliezen in de orde van grootte van 20 procent. En u raadt mogelijk al waar die energie blijft: warmteontwikkeling in de transformator. Riskeer geen brand, neem een trafo die voldoende capaciteit heeft. En als de transformator echt gloeiend heet wordt – zo heet dat u uw hand er niet even op kan houden – vervang deze dan door een zwaardere.

Conclusie

Een heel bijzonder printertje. Erg mooi, om niet te zeggen fantastisch. Het papiertransport werkt feilloos, de printer maakt geen lawaai, bovendien kan men zowel met gewoon papier als met thermisch papier werken. En burengerucht is er absoluut niet bij.

Het aantal verschillende karakters die deze printer aan boord heeft is verbluffend: zeven Romeinse alfabetten, Grieks, Russisch, Chinees, symbolen en een volledige Japanse set. Natuurlijk zijn ook de grafische MSX tekens aanwezig. En dat in een prima kwaliteit.

De printer is aan te sluiten op 220 Volt met een meegeleverde adapter, er wordt een half jaar garantie geboden middels een Nederlands garantiebewijs.
Bij de printer worden één zwart en één kleurenlint meegeleverd, nieuwe lintjes kosten ongeveer *f* 10,- voor een zwart en *f* 15,- voor een kleurenlint. Niet echt duur, maar de printer is waarschijnlijk wel een lintenvreter. In de alledaagse praktijk moet men niet al te snel even een screendumpje in kleur afdrukken.

Echt goedkoop is deze printer niet, maar gezien de nieuwe toepassingsmogelijkheden is de prijs niet overdreven. Kleurenprinters zijn nu eenmaal duur. En die kleurmogelijkheid zal toch de voornaamste reden zijn om een dergelijke technisch wondertje te bestellen. Inderdaad, bestellen, want men houdt deze machine niet op voorraad.

Wie even snel de listings wil kunnen afdrukken is beter uit met een gewone matrixprinter.
Maar voor hen die het grafisch werk belangrijk vinden en nu eindelijk in staat willen zijn om eens snel en zonder poespas een papieren kopie van hun creaties willen hebben is dit werkelijk een uitkomst.

Panasonic FS PC1, MSX 48 dots kleurenprinter
Prijs: *f* 1275,-

Verdere informatie:

MSX Centrum
Witte de Withstraat 27
1057 XG Amsterdam
Tel.: 020-167058 – na 14.00 uur

Kort Nieuws

Open Dag Enschede

De MSX Computerclub Enschede organiseert op zaterdag 19 mei een grote Open Dag. Er zullen demonstraties op zowel MSX1, MSX2 en MSX2+ gehouden worden. Verder stelt men zich ten doel iedere vraag die men maar op MSX-gebied heeft te kunnen beantwoorden. Dat klinkt wel wat pretentieus, maar we mogen aannemen dat er een redelijke hoeveelheid kennis van zaken in huis is.
De toegang voor deze dag is **gratis**. Vanaf het station is de open dag te bereiken met buslijn 7.

Datum: 19 mei 1990
Tijd: 10.00 tot 17.00 uur
Plaats: Clubgebouw 'het Bruggert'
Adres: Peter de Molijnstraat 20 (hoek Bruggertstraat)
Informatie: Jan van der Wal, tel. 053-330554

Landelijke clubdag Alphen

Op dezelfde datum vindt in een heel andere hoek van het land nog een grote clubdag plaats. De MSX-club 'Rijnstreek' organiseert deze ontmoetingsdag in Alphen aan de Rijn. Diverse andere gebruikersgroepen zullen hier ook acte de présence geven. Het thema van de bijeenkomst luidt dan ook: 'contacten tussen Nederlandse MSX clubs.
De toegangsprijs bedraagt *f* 2,50.

Datum: 19 mei 1990
Tijd: 10.00 tot 17.00 uur
Plaats: gebouw 'de Bron', naast winkelcentrum Ridderhof
Informatie: Harrie Dechering, tel. 01720-43122

Zandvoort

Op de valreep bereikte ons het bericht dat men ook in Zandvoort bezig is met het opzetten van een MSX-dag. Deze zal moeten gaan plaatsvinden op 22 september, in het gemeenschapshuis in Zandvoort.

Wie inlichtingen wil over deelname etcetera kan contact opnemen met:
Jaap Hogendijk, telefoon 02507-17966.

```
70 Kleur  15,1,1: Scherm  1,2: Schoon scherm
80 A$="MSX/MS-DOS Vraagbaak"
90 Voor  A=Lengte van (A$) Tot  1 Stap -1
100 Voor  B= 1 Tot  25-Lengte van (A$)+A
110 Positioneer cursor op  B,15
120 Schrijf " "; Midden van (A$,A,1)
130 Volgende  B: Volgende  A
140 B$=Invoer $(1)
```

NL-MSX, voorbeeld van een vertaalde listing

MS-BBS

Het BBS van Jos de Boer uit Warns — al vaker genoemd — loopt als een trein. Dat is niet vreemd, want dit BBS heeft vooral voor MSX'ers veel leuke dingen te bieden. In nummer 39 zullen wij het jongste product bespreken: MSX-Connect. Veel kleinere programma's zijn ook de moeite waard en daar willen we af en toe ook aandacht aan besteden.
Dit keer willen we wijzen op NL-MSX, een programmaatje dat Basic listings in het Nederlands vertaalt. Het werd ons toegestuurd naar aanleiding van het verhaal 'Binnen de Bytes' in MCM nummer 36. Bijgaande figuur geeft enig idee van hoe een vertaalde listing er uit gaat zien. Op het eerste gezicht is het heel vreemd, maar soms kan het inderdaad zeer verhelderend werken, vooral voor beginnende programmeurs. Ook als je zelf listings intypt kan zo'n vertaalde weergave een ware eye opener zijn. Overigens wordt niet alleen de listing vertaald, maar ook de foutmeldingen. Daarbij heeft Ries Vriend — de programmeur — nog als extra hulp tips ingebouwd met suggesties waar je de fout zou kunnen vinden.
Het enige waar je wel goed op moet letten is dat het invoeren van listingregels en Basic opdrachten wel in het Engels moet gebeuren. Even wijzigen in een vertaalde regel is dus niet mogelijk. Gelukkig kan er met een druk op F3 of Shift- F3 geswitcht worden tussen Engels en Nederlands.

Programma: NL-MSX
Leverancier: MS-BBS
Telefoon: 05149-1837
On line: 22.00-7.30 uur
Prijs: *f* 5,-

C.U.C.-Koopbeurs

Het C.U.C. journaal organiseert op 9 juni een grote Computer Koopbeurs voor MSX en PC. Deze dag gaat plaats vinden in de Groenoordhallen in Leiden. Dit wordt duidelijk een van de grotere gebeurtenissen voor hen die op koopjes uit zijn. Vele bedrijven zullen hier aanwezig zijn om hun producten te slijten, want het is al vaker gebleken dat de combinatie van MSX en PC het op beurzen goed doet. In tijdschriften is dat minder het geval, daar weten wij alles van.
In ieder geval zal MCM ook hier aanwezig zijn, met de complete set — voor zover niet uitverkocht — losse nummers en ongetwijfeld een aantrekkelijk beursaanbod. Wat dat wordt houden wij nog even voor ons.

Wie wil komen kijken, met de bon op deze pagina krijgt men *f* 2,50 korting op de entreeprijs.

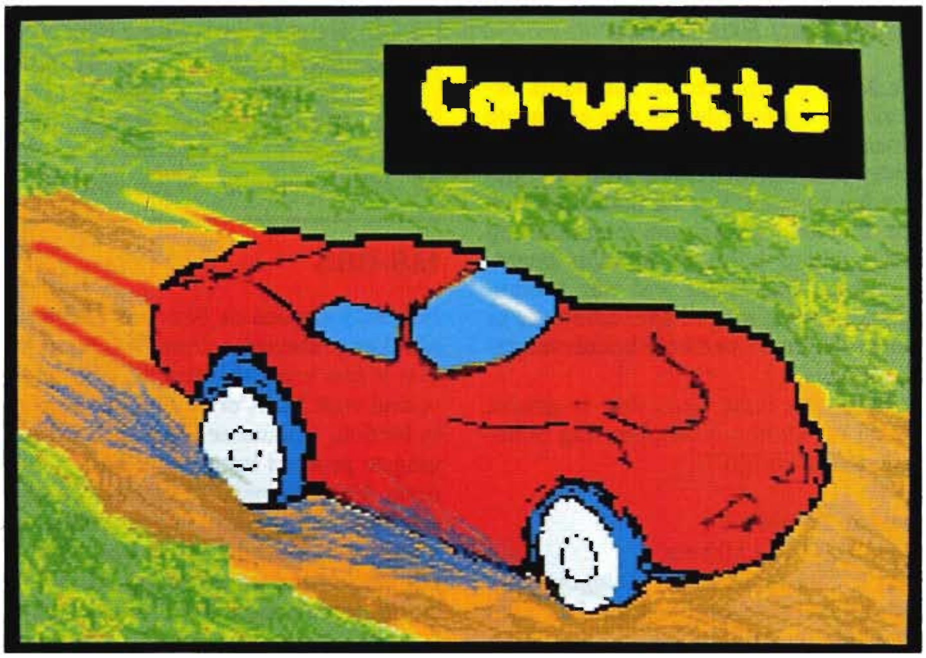
Datum: 19 juni 1990
Tijd: 10.30 — 21.00 uur
Plaats: Groenoordhallen
Adres: Willem de Zwijgerlaan 2,
2316 GB Leiden
Telefoon: 071 — 212521
Toegang: *f* 7.50; kinderen *f* 5.-

groenoordhallen leiden
9 JUNI '90
NATIONALE
PC & MSX
COMPUTER
KOOP - BEURS 90
Geopend: 10.30 - 21.00 uur
Toegang: f 7,50 volw. / f 5,-- kind

BIJ INLEVERING VAN DEZE BON KRIJGT U *f* 2,50 KORTING

MCM's Art Gallery

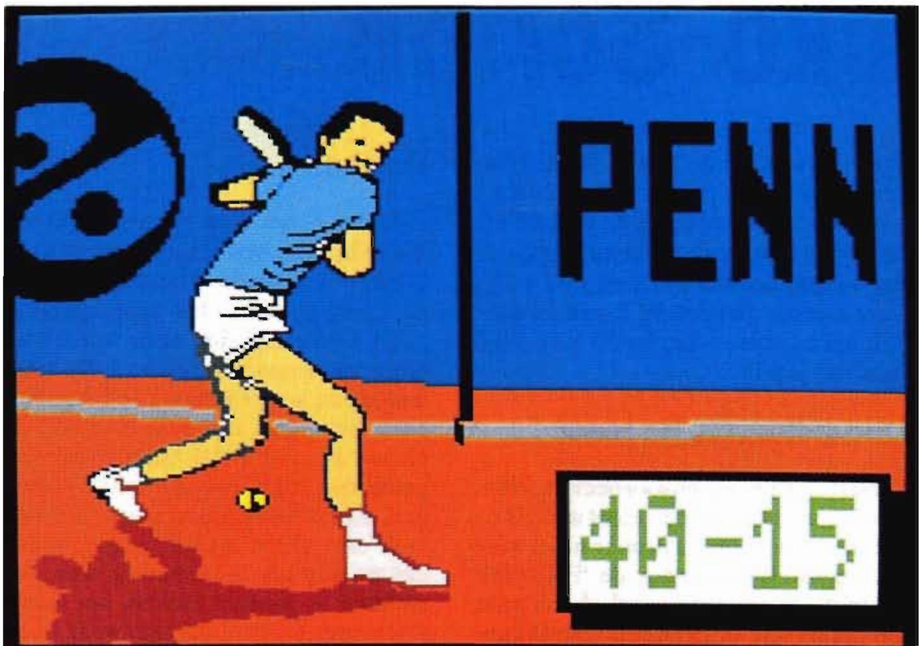
Op de vraag wat men zoals thuis met de computer doet kunnen we zo langzaam maar zeker wel een antwoord geven. In ieder geval tekenen, als we zo kijken naar de reacties op de Art Gallery. Er komen de nodige diskjes binnen, met fraaie – en minder geslaagde – prenten. Sommigen sturen hele series, die soms hartverscheurende beslissingen nodig maken.



Alle tekeningen op deze pagina's zijn gemaakt door een en dezelfde persoon, namelijk van Peter Paul Hoogbruin uit Mijdrecht. Peter Paul zond ons een hele serie tekeningen, zodat we konden kiezen. Om te beginnen rijdt een leuke rode Corvette het beeld in. Dit eigenwijze automobiel sprak ons wel aan door zijn vorm en plaatsing in het vlak. Verder vinden we de manier van tekenen heel aardig, het lijkt bijna of er met vetkrijt is gewerkt. Door de manier van tekenen en de compositie wordt er beweging gesuggereerd. Ruud Gullit is het volgend onderwerp. Ook in deze tekening zit beweging, je kunt bijna de bal die net gekopt is zien wegspringen. We willen alleen wel even iets kwijt over de schaduwen op Ruud's gezicht en arm, in deze tekening. Naar ons gevoel had er wat meer paars in gemogen, de complementaire – tegenovergestelde – kleur van geel-bruin. Dat had zeker een natuurlijker effect opgeleverd.



SCHERMKUNST OP MSX



MCM's Art Gallery staat open voor onze lezers. Zowel amateurs als professionele kunstenaars zijn welkom, en de onderwerpen zijn natuurlijk helemaal vrij. Figuratief of abstract, als het maar op MSX is gemaakt.

Stuur uw schermbeelden in op disk, onder vermelding van alle gegevens, zoals computer, video-mode en gebruikte technieken en programma's. De redactie maakt een keuze uit de inzendingen.

Wie prijs stelt op terugzending van de diskette dient een voldoende gefrankeerde antwoord-enveloppe voorzien van naam en adres bij te sluiten.

Sport

We gaan door met sport, de tennisser is ook een staaltje van mooi tekenwerk. Door deze achtergrond te kiezen is een natuurlijke omgeving ontstaan en de tennisser lijkt tegenover ons in het veld te staan.
Het portret van het meisje vinden we heel sterk in z'n eenvoud, goed kleurgebruik en weer beweging!
Als laatste een toekomstbeeld, leuk dat er voor 'vogelvlucht' perspectief is gekozen, door helder kleurgebruik is het een frisse tekening.





Techniek

Tenslotte nog wat technische gegevens. Peter Paul bezit een Philips VG-8230. Het meisje — en we nemen aan de andere tekeningen ook — zijn getekend met Designer en de joystick. Waar de onderwerpen vandaan zijn gekomen en hoe de tekeningen tot stand kwamen weten we helaas niet.
We vragen nogmaals met klem om zoveel mogelijk informatie te vermelden bij tekeningen die worden opgestuurd voor deze rubriek.

Rest ons om eenieder te bedanken voor de inzendingen. Blijf insturen, want we weten zeker dat deze kleurige rubriek wordt gewaardeerd door de lezers.
Alleen, liefst wel kant en klaar op schijf. De listing die we laatst ontvingen hebben we toch maar voorzichtig ter zijde gelegd. Alle disks worden teruggezonden, maar aan intikken beginnen we niet.

Bug in Aacko-Scribe

Zelfs de beste programma's willen wel eens een steekje laten vallen. Daarbij is het vaak vreselijk lastig om te ontdekken wat er nu precies fout gaat, laat staan om het nog verbeterd te krijgen. Vooral als het programma in kwestie – zoals AackoScribe – officieel al lang niet meer in de handel verkrijgbaar is.
Onze redactie staat echter voor niets, bij deze bieden we u althans een deeloplossing voor de probleempjes die Scribe nog steeds veroorzaakt. Veel plezier met uw vernieuwde Scribe!

PROBLEEM MINDER OP MSX

Scribe is één van de betere MSX-tekstverwerkers. Het programma is snel en biedt veel mogelijkheden. Bovendien, het werkt volgens dezelfde principes als het beroemde WordPerfect op de PC. Veel mensen die geen MSX printer hebben weten echter dat Scribe niet altijd doet wat je wilt.

Onderstrepen, ho maar

Omdat Scribe om bijvoorbeeld teksten te kunnen onderstrepen moet weten hoe 'onderstrepen' op de printer moet worden ingesteld, zit onder de 'PRINTR' functietoets een groot invulscherm waar alle codes voor de printer ingesteld kunnen worden: NLQ, vet gedrukt, onderstrepen, dubbele breedte, en zo nog wat.

Die instellingen kunnen ook op disk gezet worden, na het starten laadt Scribe automatisch de instellingen uit 'PRINTER.PRC' in. Het werkt allemaal prachtig.

Op één ding na: als zo'n printercode op een nul eindigd wordt die nul niet naar de printer gestuurd. Dat komt omdat Scribe intern een nul gebruikt om het einde van een code aan te geven. Nu zijn er printers waar onderstrepen uitgezet moet worden door een code die op nul eindigd. En dan kun je in Scribe onderstrepen wel aan zetten, maar niet meer uit...

Kortom: niet wat we willen. Ook die laatste nul moet verzonden worden. Om dat voor elkaar te krijgen - zo heeft één onzer redacteuren ondekt na een twee halve dagen durend gevecht met Scribe - moeten er wat bitjes in Scribe van één op nul gezet worden. Of voor de ML programmeurs: een JR NZ moet vervangen worden door NOP's.

SCRPCH

SCRPCH – kort voor SCRibe PatCH – is een Basic programma dat deze aanpassing voor u verzorgd. Het veranderd één van de files van Scribe zodanig dat voor elke printercodes alle drie de bytes verzonden worden. Als een code korter is staan de overige bytes toch op nul. Na het runnen van SCRPCH kan het onderstrepen tenminste weer uitgezet worden...

Patchen

Patchen – om eens een vreselijke verbastering van een Engels begrip te gebruiken – is een heel bekende techniek in computerland. De hier getoonde patch is tamelijk simpel; er worden wat bitjes van waarde veranderd. De totale lengte van het programma blijft gelijk. Maar een patch kan ook een stuk complexer zijn. Stel dat we middels een patch een extra routine willen toevoegen, dan zullen we domweg een sprong naar een nieuw stukje programma – dat we aan het al bestaande stuk vastplakken – moeten maken, om daarna weer terug te springen naar waar we gebleven waren. In zo'n geval worden dan wat instructies overschreven met de sprongopdracht, waarna we die overschreven ML-opdrachten in de uiteindelijke patch als eerste alsnog zullen opnemen.

Patchen doen we natuurlijk alleen maar als we de oorspronkelijke assemblersource niet tot onze beschikking hebben. Als er alleen maar een reeds geassembleerd ML-programma is, dan is een patch - letterlijk is dat zoiets als 'plakker' – de enige mogelijkheid. Maar de structuur wordt er allesbehalve helderder door, zeker als er in de loop der tijden meerdere complexe patches in een programma worden aangebracht.

Truukje

Voor de liefhebbers, het hier gebruikte patch-programma is een beetje bijzonder qua structuur. Wie even naar de listing kijkt, zal het opvallen dat we het SCRIBE-bestand als random-file openen. Een simpel truukje, waardoor het eigenlijke patchen heel makkelijk wordt. Sterker nog, zonder die truuk is het bijna onmogelijk om in Basic een patch te maken.

Voor Basic is namelijk CHR$(26) – de Control-Z – een einde-bestandsvlag. We zullen Basic er met geen mogelijkheid van kunnen overtuigen om bij een gewoon sequentieel bestand over die Control-Z heen te lezen. En in een ML-programma kan zo'n Control-Z best voorkomen, als instructie of als deel van een waarde.

Zodra we een bestand echter als random openen, dan zal Basic zonder enig probleem wel Control-Z lezen. Bij een random-bestand geldt de totale filelengte, zonder dat er verder einde-bestands tekens in voor kunnen komen. Vandaar dat SCRPCH het programma random opent, om daarna in één veld een stuk code met de gewraakte bytes in te lezen. Dan nog even een korte controle om te zien of de patch misschien al eens gedaan was, waarna we ook nog eens kijken of die vijf bytes wel de verwachte waarden bezitten. Immers, als we een andere versie met de verkeerde patch te lijf zouden gaan, dan is die Scribe wel met de muziek mee!

Veel plezier, met deze techniek. En wie nog meer patches bedenkt, we houden ons aanbevolen

```
1Ø REM SCRPCH, Patch voor AACKOSCRIBE
, by RWL                                    Ø
2Ø REM                                      Ø
3Ø REM MSX/MS-DOS Computer Magazine         Ø
4Ø REM                                      Ø
5Ø REM v 2.ØØ, feb '89 (Prt codes -Ø)       Ø
6Ø REM                                      Ø
7Ø PRINT "Doe Scribe disk in drive"       178
8Ø PRINT "en druk een toets: ": I$=IN
PUT$(1)                                    11
9Ø OPEN "SCRIBE.ØØ3" AS #1 LEN=128: F
IELD #1, 128 AS SC$: GET #1,38            146
1ØØ IF MID$(SC$,76,5)=CHR$(&HØ)+CHR$(
&HØ)+CHR$(&HCD)+CHR$(&HEF)+CHR$(&H92)
 THEN PRINT "ERROR: Patch al gedaan!"
: GOTO 13Ø                                217
11Ø IF MID$(SC$,76,5)<>CHR$(&H28)+CHR
$(&H7)+CHR$(&HCD)+CHR$(&HEF)+CHR$(&H9
2) THEN PRINT "ERROR: Foute versie!":
 GOTO 13Ø                                   4
12Ø MID$(SC$,76,2)=CHR$(Ø)+CHR$(Ø): P
UT #1,38: PRINT "Patch wordt gedaan."     119
13Ø CLOSE                                 236
```

MS-DOS 2.20 handboeken

Na lang wachten hebben we dan eindelijk de officiële Programmers Reference Manuals voor MSX-DOS 2.20 mogen ontvangen. Van die boeken die het verschil maken tussen moeizaam proberen of vlijmscherp programmeren.

ONMISBARE BOEKEN VOOR DE PROGRAMMEUR

Het lezen was een verademing. Eindelijk werd het één en ander echt duidelijk. Het geheugenbeheer bijvoorbeeld, waarvoor DOS 2.20 een paar heel fraaie oplossingen blijkt te kennen. Zo kan de memory-mapper in blokken van 16kB worden benut, waarbij DOS keurig bijhoudt welke blokken in gebruik zijn. Een simpele CALL en het gebruikersprogramma krijgt een blok toegewezen, mits er nog ruimte is.

Compleet

Twee kleine – A5-formaat – boekjes van ieder zo'n 70 pagina's geven in helder Engels alle relevante informatie. Deel 1 is geheel gewijd aan de 'transient programs', zeg maar de onder DOS te gebruiken programmatuur.
Heel belangrijk is natuurlijk de bestandsafhandeling, die ingrijpend veranderd is bij de nieuwe DOS. I/O, file handles, file info blocks, file control blocks, het klinkt allemaal heel bekend maar is toch een stuk uitgebreider dan we gewend waren.

Deel 2 beschrijft uiterst gedetailleerd de MSX-DOS functie-aanroepen. Daarbij valt op dat, vergeleken met de oude DOS, het aantal functies bij benadering verdubbeld is. Veel van de nieuwe functies hebben weer te maken met de ingrijpend andere manier waarop deze DOS met bestanden omgaat. Het in gebruik nemen van subdirectories is een ingrijpende zaak.

Conclusie

De informatie waar iedereen op zit te wachten. In een klein lettertje weliswaar, maar volledig en leesbaar. Met registers en inhoudsopgaven, iets opzoeken is geen lastige speurtocht.
Naar ons idee een must voor alle serieuze programmeurs, maar ook voor diegenen die nu zonder meteen te gaan programmeren eens wat meer van de interne werking van DOS 2.20 willen weten. Geen lichte kost, maar deugdelijke techneutentaal.

MSX-DOS 2.20 Programmers Reference Manuals
Twee delen
Prijs: 29,- DM per deel
Verdere informatie en bestellingen:

H.S.H. Computervertrieb GmbH
Romberg Strasse 16
D-4715 Davensberg
Duitsland
Tel.: 09-49 2593 6168

Kort en Krachtig

Men is er inmiddels wel aan gewend, dat wij met alle plezier diverse korte listinkjes publiceren. Weinig programma, veel effect, dat is het credo van deze rubriek. Iedereen mag insturen en wij publiceren een selectie op diverse niveau's: van heel eenvoudige schermgrapjes tot complexe VDP trucs. Dat betekent ook dat wij onmogelijk alles wat binnenkomt daadwerkelijk kunnen publiceren. Laat u daardoor echter niet ontmoedigen: Is uw inzending er niet bij, zend gerust nog eens wat nieuws in. Vooral op Oneliners zijn wij dol. Een Oneliner past in zijn geheel op één regel en is dus maximaal 255 tekens lang.

KLEINE MAAR FIJNE LISTINGS, VOOR VOOR DE FIJNPROEVER

Geen mens zo lastig als onze hoofdredacteur. Menig redactielid in spé is reeds mopperend maar weer heengegaan. Dat gedonder over de listing layout! Des te wonderbaarlijker dat we ditmaal één listing met uitgesproken afwijkende vormgeving hebben die blijkbaar in zijn ogen wel genade kon vinden. Anton Klein uit Loppersum maakte het bont genoeg.

STROFZ-ABHSTR

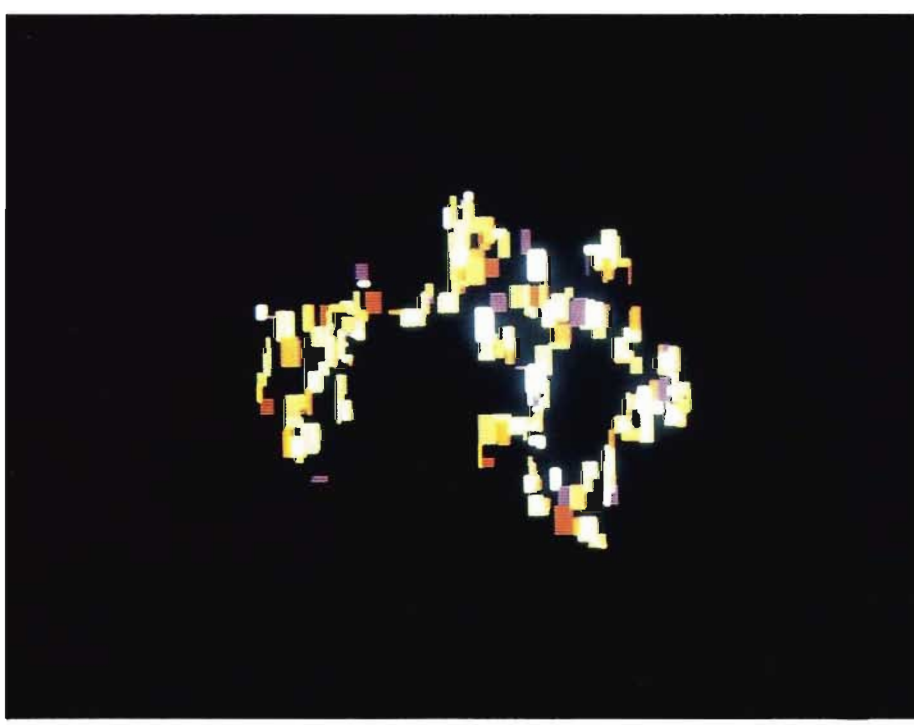
Hij stuurde ons 'Sterren ofzo', want wat het precies moest voorstellen is hemzelf blijkbaar ook niet duidelijk. Wij persoonlijk zien meer in een titel als 'Brownse beweging', hoewel ook dat de lading niet geheel dekt. Dus hebben we de oorspronkelijke titel maar gewoon afgekort: STROFZ.

Even tussendoor, die term 'Brownse beweging' is afkomstig uit de beta-vakken. Het beschrijft de volledig op toeval gebaseerde bewegingspatronen zoals bijvoorbeeld gezien kunnen worden als je melk door een microscoop bekijkt. De vetbolletjes in de melk bewegen, maar in welke richting is niet te voorspellen. Het enige dat zeker is dat — door de hele melkfles genomen — de vetbolletjes keurig verdeeld blijven in de vloeistof. In computerland wordt hetzelfde effect ook wel eens een 'random walk' genoemd.

In feite is STROFZ een heel vreemdsoortige listing: een super-oneliner — de regel is tot de nok toe vol — die barst van de COLOR opdrachten. Het resultaat is echter in het geheel niet kleurrijk te noemen. De witte vlekken dansen voor je ogen, dat is alles.

Nieuwsgierig als wij zijn, konden wij het natuurlijk niet laten om in de listing te gaan peuteren. Onze eerste actie was: eens zien wat het doet op scherm acht. In totaal werden 5 tekens gewijzigd, met een verbluffend ander resultaat. Alles behalve sterren.
Om de lezer wat meer overzicht te bieden hebben we deze variant over meerdere regels verdeeld uitgeschreven. In ABHSTR kunt u even meekijken naar wat er allemaal gebeurt.

Om maar bij het einde te beginnen, in regel 140 wordt terug gewezen naar regel 90, dit is dus de hoofdlus. In STROFZ wordt deze lus geforceerd met een valse FOR.. NEXT-lus. Valse FOR.. NEXT-lussen zijn alléén toegestaan in super-oneliners.



```
10 ' #---------------------------------#                                  Ø
2Ø ' 1 Kort maar puinzooi (krachtig ?) 1 #--------------------------#    Ø
3Ø ' 1  Sterren(of zo)    alleen MSX-2 1 1 MSX Computer Magazine     1    Ø
4Ø ' #---------------------------------# 1 KK38-1                    1    Ø
5Ø '          #-Anton Klein--------#     #--------------------------#    Ø
6Ø '          1 Wirdumerweg 35     1                                      Ø
7Ø '          1 9919 HB  Loppersum 1                                      Ø
8Ø '          #-tel. Ø5967-2156----#                                      Ø
9Ø '                                                                      Ø
1ØØ A=RND(-TIME):COLOR15,Ø,Ø:SCREEN7:CN=Ø:AC=15:X=255:Y=1Ø6:PSET(X,Y),1:FORI=ØTO
1STEPØ:CN=(CN+1)MODAC:COLOR=(CN+1,Ø,Ø,Ø):COLOR=((CN+1)MODAC+1,7,7,7):RX=RND(1)*1
8-9:RY=RND(1)*18-9:X=X+RX-(X<Ø)*9+(X>511)*9:Y=Y+RY-(Y<Ø)*9+(Y>211)*9:LINE-(X,Y),
CN+1,BF:NEXTI                                                           144
11Ø '    #------------------------------------#                           Ø
12Ø '     1 P.S.                              1                           Ø
13Ø '     1        Gefeliciteerd met back to MSX 1                        Ø
14Ø '    #------------------------------------#                           Ø
```

```
1Ø ' ABHSTR                                                               Ø
2Ø '                                                                      Ø
3Ø ' MSX Computer Magazine KK38-1b                                        Ø
4Ø ' Bewerking van Strofz                                                 Ø
5Ø '                                                                      Ø
6Ø A=RND(-TIME): COLOR 15,Ø,Ø: SCREEN 8                                  28
7Ø CN=Ø: AC=15: X=125: Y=1Ø6                                            1Ø4
8Ø PSET (X,Y),1                                                           5
9Ø CN=(CN+1)MOD AC                                                      218
1ØØ COLOR=(CN+1,Ø,Ø,Ø): COLOR=((CN+1)MOD AC+1,7,7,7)                     87
11Ø RX=RND(1)*18-9:            RY=RND(1)*18-9                           169
12Ø X=X+RX-(X<Ø)*9+(X>255)*9: Y=Y+RY-(Y<Ø)*9+(Y>211)*9                  219
13Ø LINE -(X,Y),CN+1,BF                                                 164
14Ø GOTO 9Ø                                                              2Ø
```

Vóór regel 90 vinden de diverse initialiseringen plaats: de random generator wordt gestart, het scherm ingesteld, variabelen krijgen een beginwaarde en het eerste punt wordt gezet, in zwart.
Dan gaan we de hoofdlus in. In regel 90 wordt de waarde van CN met 1 opgehoogd, tot een maximum van 14 (de waarde van AC-1). Dankzij de MOD functie beginnen we daarna weer bij nul. Regel honderd bewaren we even voor straks.

In regel honderdtien wordt voor de x en de y een verschuiving bepaald. Deze verschuiving ligt ergens tussen -9 en +9. Regel 120 bepaalt op elegante wijze de nieuwe x en y: behalve dat de verschuiving er bij opgeteld wordt, wordt middels een slimme Boole'se constructie de rand van het scherm in acht genomen. Geen zogenaamde wrap around, maar stilstaan zodra de rand bereikt wordt. Normaal gesproken zijn daar vier IF.. THEN's voor nodig; een ramp voor wie op één regel wil blijven. Regel 120 is de moeite van het onthouden waard.
Tot slot wordt er in regel 130 een blokje getekend, vanaf het vorige punt, met kleurnummer CN+1. In ABHSTR zie je dat ook vrij letterlijk gebeuren. Er ontwikkeld zich een soort slang van kleurige blokjes en al kronkelend wordt langzamerhand het hele scherm gevuld. Je begint je langzamerhand af te vragen waar al die dansende lichtjes in STROFZ vandaan komen.

Het geheim zit 'm natuurlijk in regel honderd. In ABHSTR heeft regel honderd in het geheel geen functie. Je kunt 'm net zo goed weg laten, want op scherm acht werkt de COLOR= opdracht niet. Dat komt omdat op scherm acht ieder puntje ingekleurd kan worden met één van de 256 kleuren. Voor elk puntje wordt dus een eigen kleurnummer vastgelegd.
Op andere schermen werkt dat iets anders. Meestal kan men slechts 16 kleuren benutten, genummerd van 0 tot 15. Op scherm 6 zijn dat er zelfs maar 4. Met behulp van COLOR= kan men nu zelf een kleur 'mengen' – opbouwen uit rood, groen en blauw – en deze kleur koppelen aan een bepaald nummer. Bijvoorbeeld:

COLOR=(4,6,3,0)

betekent dat aan 'palet' nummer 4 de kleuropbouw 6,3,0 wordt gebonden. Daarbij staan de getallen 6, 3 en 0 voor de intensiteit van de respectievelijk rode, groene en blauwe component. Die intensiteit mag variëren tussen 0 en 7. Reeds getekende pixels worden acuut van kleur veranderd als een COLOR= zich voordoet.

```
1Ø ' POTLOD                                   Ø
2Ø '                                          Ø
3Ø ' MSX Computer Magazine                    Ø
4Ø ' KK38-2                                   Ø
5Ø ' ingezonden door Erik Wannee,             Ø
6Ø ' Rozendaal                                Ø
7Ø '                                          Ø
8Ø POKE &HFBB1,1: J=15: COLOR J,1,8:
SCREEN 2,1: SOUND 6,21: SOUND 7,42: S
OUND 8,5                                      38
9Ø SPRITE$(1)=CHR$(192)+CHR$(224)+CHR
$(112)+CHR$(56)+CHR$(28)+CHR$(14)+CHR
$(7)+CHR$(2): PSET (1ØØ,1ØØ)                  138
1ØØ A=RND(-TIME)*12Ø-6Ø: IF X+A>256 O
R X+A<Ø THEN GOTO 1ØØ                         26
11Ø X=X+A                                     21Ø
12Ø A=RND(1)*12Ø-6Ø: IF Y+A>19Ø OR Y+
A<Ø THEN GOTO 12Ø                             128
13Ø Y=Y+A: LINE -(X,Y): PUT SPRITE 1,
(X,Y),J,1: SOUND Ø,Y+5Ø: I=I+1: IF I<
1ØØ THEN GOTO 1ØØ                             79
14Ø IF X>3Ø THEN PAINT (X-13,Y),J             14
15Ø I=Ø: J=J-11: IF J<Ø THEN J=15: CO
LOR 1: LINE (Ø,Ø)-(255,191): PAINT (1
Ø,Ø): PAINT (1Ø,191): FOR T=1 TO 3ØØ:
 NEXT T: CLS                                  229
16Ø COLOR J: PUT SPRITE 1,(X,Y),J,1:
GOTO 1ØØ                                      183
```

```
1Ø ' TEKST                                    Ø
2Ø '                                          Ø
3Ø ' MSX Computer Magazine                    Ø
4Ø ' KK38-3                                   Ø
5Ø ' ingezonden door P. Hoogbruin,            Ø
6Ø ' Mijdrecht                                Ø
7Ø '                                          Ø
8Ø ' Machinecode in geheugen poken **         Ø
9Ø RESTORE 2ØØ                                172
1ØØ FOR X=&H9ØØØ TO &H9Ø26                    152
11Ø READ A$: POKE X,VAL("&H"+A$)              125
12Ø NEXT X                                    97
13Ø ' Hookadres op &H9ØØØ stellen ***         Ø
14Ø POKE &HFDC3,Ø                             79
15Ø POKE &HFDC4,&H9Ø                          193
16Ø POKE &HFDC2,&HCD                          243
17Ø REM Zet hook weer op standaard wa
arde met POKE &HFDC2,&HC9 ***********         Ø
18Ø CLS                                       13
19Ø END                                       188
2ØØ DATA 2A,DC,F3:' LD HL,(F3DCH)             74
21Ø DATA E5       :' PUSH HL                  2Ø3
22Ø DATA 26,13    :' LD H,19
; kolompositie karakterstring                 91
23Ø DATA 2E,ØA    :' LD L,1Ø
; rijpositie   karakterstring                 57
24Ø DATA CD,C6,ØØ:' CALL POSIT                249
25Ø DATA 11,22,9Ø:' LD DE,data
; positie karakterstring  in  geheuge
n                                             45
26Ø DATA Ø1,Ø5,ØØ:' LD BC,5
; grootte karakterstring incl. spatie
s                                             217
27Ø DATA 78       :' loop; LD A,B             13
28Ø DATA B1       :' OR C                     183
29Ø DATA ØB       :' DEC BC                   216
3ØØ DATA 2Ø,Ø5    :' JR NZ,hulp               132
31Ø DATA E1       :' POP HL                   99
32Ø DATA CD,C6,ØØ:' CALL POSIT                246
33Ø DATA C9       :' RET                      149
34Ø DATA 1A       :' hulp; LD A,(DE)          99
35Ø DATA 13       :' INC DE                   2Ø3
36Ø DATA CD,A2,ØØ:' CALL CHPUT                165
37Ø DATA 18,EF    :' JR loop                  58
38Ø DATA 2Ø,4D,53                             184
39Ø DATA 58,2Ø    :' data: twee spatie
s en de letters MSX                           128
```

Gewapend met deze kennis moet het mogelijk zijn om regel honderd te doorgronden. In het eerste deel wordt palet nummer CN + 1 vastgesteld op 0,0,0; oftewel zwart. In regel 130 wordt het blokje in kleurnummer CN + 1 getekend, het is dus even zwart als de achtergrond zelf! In de tweede helft van regel krijgt het palet ((CN + 1)MOD AC) + 1 de kleur 7,7,7; wit.
Aangezien CN + 1 steeds rondloopt van 1 tot 15, zal het getekende blokje na 14 keer door de lus gelopen te zijn plotseling wit oplichten en vanaf dat moment zal dat oplichten zich herhalen iedere vijftiende keer dat de lus doorlopen wordt.
Tenzij ondertussen de kop van de slang er over heen is gelopen – hetgeen wij dus niet kunnen zien. Dan is er plotseling een hoekje uit, of zijn er twee kleinere blokjes over. Dit alles wordt langzaam duidelijk als je eens wat gaat experimenteren met de waarden van de kleurintensiteiten en paletnummers.

Wij eindigden met vier COLOR = opdrachten achter elkaar, waarin alle variabelen werden benut om de boel zo bont mogelijk te maken. Toen riep de plicht.

POTLOD

Toch wel onhandig, die zesletterige programmanamen. Dat moet, omdat we dan een éénduidige aanduiding hebben, waaronder de programma's ook op de cassettes – van onze cassetteservice – terug te vinden zijn. Want zelfs de kleinste oneliner gaat mee. Goed, Erik Wannee uit Rozendaal stuurde POTLOOD in. En als uw nerven het COLOR = avontuur hebben overleefd, moet u dit ook aankunnen.

Het nijvere potloodje bekrast zingend uw scherm, en niet zo'n beetje. Twee kleuren over elkaar en op scherm 2 geeft dat wel een lichte color spill – stukjes van de reeds getrokken lijn nemen de kleur over van de latere lijn, maar in dit geval zal niemand daaronder lijden.

```
10 ' CUJOMU                                                              0
20 ' MSX Computer Magazine KK28-4                                        0
30 ' ingezonden door Herman Post, Enschede                               0
40 '                                                                     0
50 DEFINT A-Z: SCREEN 5: DIM JC(8,1): COLOR 15,4,4                      38
60 H=211: B=256: LX=B\2: LY=H\2                                        113
70 SPRITE$(0)=CHR$(16)+CHR$(16)+CHR$(16)+CHR$(254)+CHR$(16)+CHR$(16)+CHR$(16)   223
80 ' even kijken of er nul, één of twee knaagdieren zijn aangesloten ***********   0
90 ' poort één:                                                          0
100 S=0: FOR I=1 TO 10: P=PAD(12): S=S+PAD(13)+PAD(14): NEXT I         252
110 IF S<>0 AND S<>20 THEN GOTO 100                                    116
120 IF S=0 THEN M=1 ELSE M=0                                           204
130 'poort twee:                                                         0
140 S=0: FOR I=1 TO 10: P=PAD(16): S=S+PAD(17)+PAD(18): NEXT I          12
150 IF S<>0 AND S<>20 THEN GOTO 140                                      4
160 IF S=0 THEN M=M+2                                                   31
170 ' afhankelijk van het aantal muisjes de twee functie's inlezen. ************   0
180 IF M=0 THEN DEF FN VR=JC(STICK(1),1)+JC(STICK(2),1)+JC(STICK(0),1):
                DEF FN HR=JC(STICK(1),0)+JC(STICK(2),0)+JC(STICK(0),0)  187
190 IF M=1 THEN DEF FN VR=PAD(12)+PAD(14)+JC(STICK(2),1)+JC(STICK(0),1)+1:
                DEF FN HR=PAD(12)+PAD(13)+JC(STICK(2),0)+JC(STICK(0),0)+1   22
200 IF M=2 THEN DEF FN VR=PAD(16)+PAD(18)+JC(STICK(1),1)+JC(STICK(0),1)+1:
                DEF FN HR=PAD(16)+PAD(17)+JC(STICK(1),0)+JC(STICK(0),0)+1   23
210 IF M=3 THEN DEF FN VR=PAD(12)+PAD(14)+PAD(16)+PAD(18)+JC(STICK(0),1)+2:
                DEF FN HR=PAD(12)+PAD(13)+PAD(16)+PAD(17)+JC(STICK(0),0)+2   107
220 FOR I=0 TO 8: FOR J=0 TO 1: READ JC(I,J): NEXT J,I                 186
230 DATA 0,0,0,-1,1,-1,1,0,1,1,0,1,-1,1,-1,0,-1,-1                     240
240 ' Hoofdlus ************************************************************   0
250 GOSUB 290                                                          170
260 PUT SPRITE 0,(LX-3,LY-3),1,0                                         4
270 GOTO 250                                                            76
280 ' Subroutine joystick/cursor/muis ***************************************   0
290 LY=(LY+FN VR+H)MODH                                                 29
300 LX=(LX+FN HR+B)MODB                                                 52
310 RETURN                                                             191
```

Erik verzocht ons uitdrukkelijk niet aan zijn listing te knoeien. We hebben ons dus ingehouden, alleen spaties achten wij nog steeds onontbeerlijk voor hen die weinig type- en programmeer ervaring hebben. Verder hebben wij de sprite$ helemaal uit CHR$ opgebouwd, anders krijgt onze listingprinter problemen. Wel willen we nog wat toelichting kwijt, voor wie thuis wil knutselen:

De POKE aan het begin van regel 80 schakelt de Ctrl-Stop uit. Dat is handig als je niet wilt dat een willekeurige onbenul het programmaatje onderbreekt — op een demonstratie of zo — maar uiterst irritant als je aan het experimenteren bent. Veertien keer DEL en één keer RETURN doen wonderen.
Het is een kwestie van smaak, maar in onze ogen is de rode border wat veel van het goede. Persoonlijk zetten wij in de COLOR opdracht in regel tachtig liever een één dan een acht. Verder kan een wachtlus in regel 130 voor sommige mensen nuttig zijn om het potloodje überhaupt te kunnen zien. Dankzij alle spaties is het geheel al minder zenuwslopend dan het was, zodat wij geen extra wachttijd hebben ingebouwd. MSX2'ers tenslotte kunnen natuurlijk de color spill vermijden door op scherm vijf te werken.

TEKST

Peter Paul Hoogbruin uit Mijdrecht stuurde ons een aardig machinetaal programmaatje dat telkens als de computer op invoer staat te wachten een tekst op het scherm afdrukt. Vermoedelijk een uiterst leerzaam listinkje voor beginnende machinetaalprogrammeurs.
De machinetaal wordt bereikt door gebruik te maken van de Hook H.CHGE, die wordt bereikt vanuit de BIOS routine CHGET. Iedere keer als CHGET wordt aangeroepen — waarover straks meer — wordt er even naar H.CHGE op adres &HFDC2 gekeken, of daar soms iets te doen is. Normaal gesproken staat daar alleen de machinetaal code voor RET van return: &HC9; waarna CHGET zijn gewone werk doet: wachten op een toets. In regel 160-140, van boven naar onder dus, wordt er in deze hook een verwijzing gezet: voer eerst de opdrachten vanaf &H9000 uit. En vanaf &H9000 plonzen we dus in de ML. Wie werkelijk het naadje van de kous wil weten, kan het beste op een donderdagavond tussen zeven en half negen naar de vragenlijn bellen en dan aan Markus The — het beruchte hoofd programmaredactie — om uitleg vragen. In dit korte bestek beperken wij ons tot één addertje dat zich onder het gras schuil houdt. In regel 260 wordt namelijk ver-

```
10 ' STORIN                                              0
20 ' MSX Computer Magazine KK28-5                        0
30 ' ingezonden door Robert van Veldhoven                0
40 '                                                     0
50 COLOR 15,1,1: SCREEN 3                              189
60 OPEN "grp:" AS #1: PRESET (25,70)                    30
70 PRINT #1,"STORING"                                  198
80 VDP(10)=VDP(10) XOR 32                              180
90 GOTO 80                                             153
```

meld dat hier de lengte van de tekst wordt opgegeven, in dit voorbeeld vijf tekens. Inderdaad wordt in register BC 05 gezet en bevatten de laatste twee regels vijf data-elementen: spatie, M, S, X, spatie. Wie nu P. Hoogbruin wil afdrukken moet dus — met spaties aan het begin en eind mee — 14 tekens plaatsen. Dus de 05 in regel 260 wordt 0E en vanaf regel 380 worden de hexadecimale karaktercodes van P. Hoogbruin opgenomen. Maar daarmee zijn we er nog niet, want we hebben extra DATA toegevoegd. Dus bij het inlezen van de DATA vanuit Basic moet het eindadres ook aangepast worden: regel 100. Voor ons voorbeeld moet de eindwaarde van de teller X op &H902F gezet worden. Begrijpt u nu waarom wij zo dol zijn op asterisken — zoals het sterretje officieel nu eenmaal heet — om het einde van DATA te markeren?

Tot slot nog even het aanroepen van CHGET. CHGET wordt constant gebruikt als je in de Basic editor bezig bent, maar ook tijdens het lopen van programma's, om op invoer te wachten bij de diverse INPUT opdrachten. Het enige wat ontbreekt is de mogelijkheid om de tekst weer te wissen. INKEY$ is uitdrukkelijk een ander verhaal: voor INKEY$ wordt niet gewàcht op invoer, en dus wordt de tekst uit TEKST niet vertoond. Wij hebben zo gauw geen nuttige toepassing voor dit alles kunnen verzinnen, maar die openbaart zich ongetwijfeld op een dag.

CUJOMU

Op de valreep viel dan toch de aanvulling op de cursor/joystick/muis routine van Herman Post in de bus. Dat wil zeggen, in de tussentijd was het geheel herschreven en er alleen maar beter op geworden. CUJOMU kijkt zelf of er één dan wel meer muizen zijn aangesloten en zo ja, op welke poort. Vervolgens worden op grond hiervan twee functies aangemaakt die alle mogelijke bewegingen door muis, cursor of joystick combineren tot één verandering in de x-richting, en één in de y-richting. Feilloos. Opvallend is de wijze waarop de aanwezigheid van de muis wordt onderzocht: Herman maakt gebruik van het feit dat de betreffende PAD's altijd de waarde 1 geven als er géén muis aanwezig is. Theoretisch moet het natuurlijk mogelijk zijn om tijdens dit onderzoek de muis zo te bewegen dat de totaal uitkomst van regel 100 of 140 ook precies 20 is, maar de kans dat dat per ongeluk gebeurt is verwaarloosbaar klein.

Wij hebben ons de vrijheid gepermitteerd om toch nog één ingreep te doen in Herman's product: dat betreft de 'wrap around'. Herman had de MOD-functie die hiervoor verantwoordelijk is reeds in de DEF FN opdrachten opgenomen. Wij achten het beter om deze stap toch pas in de hoofdlus te doen. Ten eerste bevordert dat de overzichtelijkheid. Ten tweede is het nu makkelijker om de wrap around te vervangen door het blijven hangen aan de rand van het scherm, zoals dat bij STROFZABHSTR besproken is.
Vervolgens hebben wij de maten van het scherm aan de initialisatie toegevoegd, met de variabelen H en B. Dat is handig als je op een ander scherm wilt gaan werken, je hoeft dan alleen de eerste regels te wijzigen. Tot slot dienen de vele spaties in de regels 180-210 de leesbaarheid. Neem ze gerust wel over, de snelheid van de uiteindelijke routine wordt er niet door beïnvloed.

Als kleine wederdienst willen we even de gebruikersgroep waarvan Herman voorzitter is noemen. Een kleine doch actieve groep, die eens in de twee maanden bijeen komt.
Het contact adres is:

MSX-club Enschede
p/a J. v.d. Wal
Putterstraat 1
7523 ZM Enschede
Tel.: 053-330554

Op 19 mei organiseren zij een open dag op het adres:
Peter de Molijnstraat 20 te Enschede.
Een goede gelegenheid om eens kennis te maken!

STORIN

Tot slot selecteerden we uit een rijtje nieuwe inzendingen van Robert van Veldhoven STORING. Het is een wat vreemd programma, eigenlijk volslagen onzin. Tot nu toe heeft bij ons op geen twee MSX'en hetzelfde resultaat opgeleverd.

De eerste drie regels laten zich makkelijk lezen: stel scherm drie in, open dat scherm om tekst op te kunnen afdrukken, ga naar puntje 25,70 en zet daar de tekst storing neer. De meeste mensen zullen al wel weten dat een dergelijk woord op scherm drie meteen schermbreed is. Wat hierna gebeurt is afwachten. Soms ziet het er meteen gestoord uit, soms moet je even op een toets drukken. Ook kan het tussenvoegen van enige actie — bijvoorbeeld X=X+1 — aan het begin van regel tachtig leiden tot trillingen of ander ongemak. Uw computer is zichzelf niet meer.

Wat er nu precies aan de hand is, dat ons is eerlijk gezegd ook een raadsel. Zelf het jongste redactie-whizzkid — Paul te Bokkel — doet er het zwijgen toe. Natuurlijk hebben we wel even opgezocht wat er officieel gebeurt.
Bit 5 van VDP register 9 wordt aan en uit gezet. Dit bit regelt samen met de buurman of -vrouw, bit nummer 4, de synchronisatie mode. Deze is eigenlijk alleen van nut als je gaat digitaliseren. Maar blijkbaar wordt de beeldscherm opbouw even verstoord door het aan en uitzetten van dit bit. Een MSX2+ vertoont weer een heel eigen effect!

Tokens enzo: Basic intern

Zoals halverwege de spellenrubriek in nummer 34 is uitgelegd worden Basic woorden in de vorm van tokens opgeslagen. Dat kost per keyword maar één byte. Maar een programma bestaat uit meer dan keywords alleen. Die rest wordt ook op allerlei manieren gecodeerd. Al lezend wordt de beginner gevorderde.

MSX-BASIC PROGRAMMA'S IN HET GEHEUGEN

Het geheugen van een computer bestaat uit een lange reeks bytes. In elke byte kan een getal van 0 tot en met 255 worden opgeslagen. Zoals we gezien hebben kan zo'n getal voor een heel keyword staan. Het woord PRINT bijvoorbeeld wordt voorgesteld door de code 157. Een getal in het geheugen kan echter ook allerlei andere betekenissen hebben.

Om geheugenplaatsen aan te kunnen duiden hebben ze een adres. Dat adres is niets meer of minder dan het nummer van die byte. Voor de MSX loopt de nummering van 0 tot en met 65535, een aantal dat meestal wordt aangeduid als '64K'. Dat klopt aardig als je bedenkt dat die 'K' voor 1024 staat. In het binaire rekenen van de computer is dat namelijk een mooi afgerond getal.

Het begin

Ingetikte Basic programma's komen natuurlijk ergens in dat geheugen terecht. Het kan per MSX verschillen waar dat precies is. Meestal echter zal adres 32769 het begin van het programma bevatten.
Omdat de computer zelf ook moet weten waar het programma staat is het beginadres op een vaste plaats opgeslagen. De adressen 63094 en 63095 bevatten die beginpositie. Het uitlezen van geheugenplaatsen kan met de functie PEEK(). Probeer maar eens:

```
PRINT PEEK(63094);PEEK(63095)
```

Waarschijnlijk is het resultaat iets in de trant van:

```
1 128
```

Om nu uit deze twee getallen het beginadres van het programmagebied te berekenen moet er een kleine formule op worden losgelaten. Die formule bestaat uit het vermenigvuldigen van het tweede getal met 256 en het erbij optellen van de eerste. In dit geval dus:

256*128+1=32769

Maar zoals gezegd, dat kan op uw MSX iets anders zijn. Met name als er minder dan 32K geheugen aanwezig is, of als er een cartridge geïnitialiseerd is.

Het is ook mogelijk het beginadres van een Basic programma te veranderen door – met de POKE instructie – op de adressen 63094 en 63095 de juiste waarden te zetten. Programma's die – uit Basic – de memory mapper willen gebruiken maken daar meestal handig gebruik van.
Door het programma op een andere plaats in het geheugen te zetten kan er een blok van 16K vrijgemaakt worden. Zo'n blok is nodig om het RAM dat in de mapper aanwezig is aan te kunnen spreken.

Regels

Als u ooit een Basic programma geschreven of overgetypt heeft zult u weten dat zo'n programma uit regels bestaat. Er kunnen regels worden ingevoegd, verwijderd of gewijzigd. Het is dan ook niet verwonderlijk dat een Basic tekst ook in de vorm van een serie regels in het geheugen staat. Elke regel heeft weer dezelfde opbouw:

2 bytes met het adres van de volgende regel;
2 bytes voor het regelnummer;

ruimte voor de Basic commando(s) als tokens en

1 byte die altijd 0 bevat om de regel af te sluiten.

Als de twee bytes met het adres voor de volgende regel nullen bevatten is dat voor de interpreter het teken dat het programma hier afgelopen is.
Zowel het regelnummer als het adres van de volgende regel zijn op dezelfde manier opgeslagen als het beginadres van het programma. De truuk is ook hier de tweede byte met 256 vermenigvuldigen en de eerste erbij optellen. Het regelnummer van de eerste regel in een programma is meestal dan ook te achterhalen met:

```
PRINT 256*PEEK(32772)+
PEEK(32771)
```

Deze kennis is gevangen in Linlen. Het programmaatje maakt een overzicht van de lengtes van de in het geheugen aanwezige Basic regels. Meestal zullen dat natuurlijk de regels 10 tot 60 van het

```
10 ' Linlen, MCM 1989                            0
20 '                                             0
30 AD=256*PEEK(63095!)+PEEK(63094!)            195
40 RN=256*PEEK(AD+1)+PEEK(AD+0)                  6
50 IF RN=0 THEN END                             87
60 RG=256*PEEK(AD+3)+PEEK(AD+2)                 71
70 PRINT "Regel:";RG;"is";RN-AD;"byte
s lang"                                        239
80 AD=RN: GOTO 40                               93
```

```
10 ' Linprt, MCM 1989                            0
20 '                                             0
30 AD=256*PEEK(63095!)+PEEK(63094!):
WD=PEEK(62384!)                                 55
40 RN=256*PEEK(AD+1)+PEEK(AD+0): IF R
N=0 THEN END ELSE RG=256*PEEK(AD+3)+P
EEK(AD+2)                                      245
50 PRINT: PRINT "Regel:";RG                    218
60 FOR F=0 TO RN-AD-1                          127
70  I=PEEK(AD+F): C$=CHR$(I): IF I<32
 OR I>126 THEN C$=" "                          240
80  PRINT USING "### &  ";I;C$;: IF (
F-3)MOD(WD\7)=0 AND F<>RN-AD-1 THEN P
RINT                                            74
90 NEXT F: AD=RN: GOTO 40                       45
```

programma zelf zijn, maar door het zaakje op hoge regelnummers achter een ander programma te plakken kan van elk programma een overzicht gemaakt worden van de lengtes van de regels.
Die regellengte wordt berekend door van het adres van de volgende regel het adres van de huidige af te trekken. Het resultaat is precies het aantal bytes dat deze regel in het geheugen inneemt. Dat aantal zal altijd groter zijn dan vijf omdat er naast het commando dat op die regel staat er zoveel bytes gebruikt worden voor de interne administratie.

Deze beschouwing roept waarschijnlijk de vraag op wat nu de kortst denkbare Basic programmaregel is. Welnu:

5 '

ziet er wel kort uit maar kost toch acht bytes. Ook de spatie weglaten helpt niets. Beduidend korter is:

5 REM

Kennelijk is het token voor REM korter dan dat voor het enkele aanhalingsteken. Dit lijkt ons een duidelijk hint voor mensen die met geheugen woekeren en toch commentaar in hun programma's willen opnemen!
Met zes bytes lengte is deze regel de kortst denkbare – op normale wijze in te voeren – Basic regel. Met een aantal goed gerichte POKE's is het inderdaad mogelijk een lege regel te creëren, maar dat is een stukje valsspelen van de bovenste plank.

De tokens

Iets uitgebreider is Linprt. Deze listing geeft van iedere regel het regelnummer en byte-voor-byte de inhoud. Ideaal om de zaken eens tot op de bodem uit te pluizen.
Op het eerste gezicht is er totaal geen structuur te ontdekken in de getallen die op het scherm rollen. Zeker de – toch complexe – regels van het programma zelf zijn moeilijk te herkennen. Misschien gaat het beter als we het voorbeeldprogrammaatje uit het eerdergenoemde artikel er weer bijhalen:

```
10 LET a=1
20 LET b=10
30 FOR f=a TO b
40 PRINT f
50 NEXT f
```

In figuur 1 is te zien wat er gebeurt als we Linprt op deze regels loslaten. Om dat voor elkaar te krijgen hebben we met Linprt in het geheugen het commando:

RENUM 60000

gegeven. Vervolgens kunnen we de bovenstaande regels typen en er met een instructie als:

GOTO 60000

een overzicht van op het scherm zetten. Dat overzicht staat zoals gezegd in figuur 1. De eerste vier getallen van elke regel bevatten het adres van de volgende regel en het regelnummer. Die zijn dus niet interessant als we naar de inhoud ervan willen kijken.

De werkelijke inhoud van regel 10 begint met 136. Kennelijk is dat het token voor LET. Dat klopt precies met regel 20 die ook met LET begint.
Na het LET-token volgt op beide regels een spatie. De code daarvan is 32.

Direct daarachter staat de naam van de betrokken variabele. In regel 10 een A, in regel 20 een B. Dat is duidelijk te zien omdat de beide regels voor de rest praktisch identiek zijn.
Het volgende byte bevat weer een token. Het is opvallend dat ook het symbool '=' door een token wordt vervangen. Je zou zeggen dat de ASCII code van '=' – dus decimaal 61 – gebruikt zou kunnen worden. Kennelijk dachten de makers van Basic daar anders over.

De laatste byte in elke regel bevat de waarde die aan de variabele wordt toegekend. Kennelijk kan de één in regel 10 in een enkele byte gecodeerd worden, terwijl er voor de tien uit regel 20 twee bytes nodig zijn.
Voor de representatie van getallen worden in MSX Basic verschillende methoden gebruikt. Afhankelijk van de vraag of het getal uit één of meer cijfers bestaat, of het enkele of dubbele precisie is, of het hexadecimaal of binair is opgegeven wordt het op een bepaalde manier gecodeerd.
Nu moet het mogelijk zijn de getallen van regel 30 te begrijpen. De goede vertaling is:

36 en 128: adres van de volgende regel
30 en 0: regelnummer: 256*0 + 30 = 30
130: het token voor FOR
32: een spatie
70: de variabelenaam F
239: het token voor '='
65: de variabele A
32: nog een spatie
217: het token voor TO
32: een spatie
66: eindvariabele B
0: het einde van de regel

Lastig

De complete structuur van Basic programma's is tamelijk lastig te begrijpen. Er wordt behoorlijk gegoocheld met tokens en andere coderingen. Maar als de structuur eenmaal duidelijk is kan die kennis vaak nuttig gebruikt worden. De

toepassingen lopen van het automatisch verwijderen van spaties en REM regels tot het in een variabele opgeven van het regelnummer voor een RESTORE.
Maar we zullen het voorlopig maar eenvoudig houden en de interne coderingen voorlopig laten voor wat ze zijn. Ook met de regels op zich zijn namelijk wel wat grappen en grollen uit te halen.

Wel of geen listing?

Als eenmaal duidelijk is hoe de Basic opgeslagen is kunnen er met welgemikte PEEK en POKE instructies vaak leuke effecten bereikt worden. Het commando NEW bijvoorbeeld doet niets anders dan de eerste twee bytes van het programma op nul zetten en wat systeemvariabelen aanpassen. Die twee bytes geven normaal gesproken het adres van de volgende regel, maar als ze op nul staan duidt dat op het einde van het programma.
Door nu die eerste twee bytes weer van de goede waarden te voorzien kan een verloren programma teruggehaald worden. Zelfs na een reset werkt deze truuk. Vergeet echter niet de systeemvariabelen daarmee niet op orde komen!

Als een verloren programma na twee POKE's dan ook weer te listen is moet het eerst als ASCII op disk of cassette gezet worden. Door die ASCII versie daarna weer in te laden worden de systeemvariabelen automatisch goed ingesteld. Veel eenvoudiger is het echter een OLD programma – zoals bijvoorbeeld dat uit MCM 25 – te gebruiken. Dergelijke programma's herstellen alles automatisch.
Maar er is meer te doen. Het zal u misschien opgevallen zijn dat Basic niet alle regelnummers accepteert die theoretisch mogelijk zijn. In de twee bytes die voor dat nummer worden gebruikt kan in principe een getal staan van 0 tot 65536. Maar het invoeren van

```
65535 REM test
```

levert direct een Syntax error op. Om te achterhalen waar precies de grens ligt speelden we een kort hoger-lager spelletje met de interpreter. Al snel blijkt dat regel 65529 nog net mag, terwijl 65530 een foutmelding geeft. Regel 0 – vaak ook een speciaal geval – is trouwens wel een normale regel als elke andere.

Regelnummers groter dan 65529 worden dus niet geaccepteerd. Maar met POKE instructies is het natuurlijk wel mogelijk het regelnummer van een willekeurige regel op 65530 te zetten, of nog makkelijker: op 65535. Dat laatste getal wordt voorgesteld door de waarde 255 in elke byte.

Als het eerder genoemde voorbeeldprogramma in het geheugen staat en het beginadres – meestal zal dat 32769 zijn – bekend is, dan kan het nummer van de eerste regel op 65535 gezet worden met:

```
POKE 32771,255
POKE 32772,255
```

Het resultaat is verbluffend: LIST geeft enkel nog 'Ok' op het scherm, maar na een RUN wordt het programma nog gewoon uitgevoerd. De routines die een listing op het scherm zetten verslikken zich in het hoge regelnummer en stoppen ermee. Tijdens het uitvoeren van het programma zijn de regelnummers kennelijk minder van belang.

Heel wat programmeurs proberen op deze manier hun programma voor anderen verborgen te houden, maar heel wat 'krakers' kennen de truuk en laten de listing in no-time verschijnen door het betrokken regelnummer weer op 0 te zetten. Op die manier houden twee grote groepen computeraars elkaar bezig. Mocht u trouwens toevallig tot de laatste groep behoren, juich dan niet te vroeg: er zijn oneindig veel truuks die in dit artikel niet behandeld zijn!

Figuur 1: Uitvoer van Linprt, smalle versie op MSX1

```
Regel: 10
 11    128     10     0
136     32     65 A  239     18
  0
Regel: 20
 22    128     20     0
136     32     66 B  239     15
 10      0
Regel: 30
 36 $  128     30     0
130     32     70 F  239     65 A
 32    217     32     66 B    0
Regel: 40
 44 ,  128     40 (   0
145     32     70 F   0
Regel: 50
 52 4  128     50 2   0
131     32     70 F   0
```

Figuur 1a: Brede versie, zoals die op een MSX2 verschijnt

```
Regel: 10
 11    128     10     0
136     32     65 A  239     18      0
Regel: 20
 22    128     20     0
136     32     66 B  239     15     10      0
Regel: 30
 36 $  128     30     0
130     32     70 F  239     65 A   32    217     32     66 B    0
Regel: 40
 44 ,  128     40 (   0
145     32     70 F   0
Regel: 50
 52 4  128     50 2   0
131     32     70 F   0
```

Het CMD-commando: onbekend gebied

MSX-Basic mag dan niet de snelste der programmeer talen zijn; veelzijdig is de taal in ieder geval wel. Niet alleen beschikt MSX-Basic over een heel arsenaal aan grafische en geluidsinstructies, maar er zijn ook verschillende manieren beschikbaar om machinetaal te combineren met Basic.

De meest gebruikte is de USR-functie, die we al eens uitgebreid behandeld hebben in MCM 16, in deel negen van de MSX machinetaalcursus om precies te zijn. Een andere bekende manier is via een CALL-opdracht, maar dit heeft nogal wat voeten in de aarde. Wel is het beslist de krachtigste manier om ML beschikbaar te maken onder Basic. De qua gebruik fraaiste methode echter zetten we in dit artikel eens voor het voetlicht. Overigens, de listing bij dit artikel is – ondanks het 80-koloms formaat – voor zowel MSX1 als MSX2 geschikt. Die derde methode die maar zelden gebruikt wordt is het CMD-commando. Hierover is maar weinig bekend en dat is jammer. Het gebruik van CMD is niet moeilijker dan USR, maar ook wat beperkter. Daar staat dan weer tegenover dat het veel eleganter is:

CMD COLOR

staat nu eenmaal een stuk beter – en is bovendien véél leesbaarder – dan een constructie zoals:

A=USR(16)

Onbekend maakt onbemind

De reden, dat het CMD-commando vrijwel nooit gebruikt wordt is simpel: niemand schijnt te weten hoe het precies werkt. Zelfs het alom geprezen MSX Technical Data Book – de 'MSX-bijbel' – levert hierover niet genoeg informatie. Het CMD-commando werkt via een hook en dat is in zekere zin de achilleshiel van alle MSX-documentatie: over het precieze gebruik van hooks is maar weinig bekend. Dat ligt waarschijnlijk aan Microsoft zelf, die de broodnodige gegevens van alle hooks nooit gepubliceerd heeft. Hiermee wordt het nut van een hook eigenlijk een beetje teniet gedaan... De beste bron van informatie is daarom vaak de computer zelf: de BIOS- en Basic-ROM laten immers precies zien hoe één en ander in zijn werk gaat. Helaas is het niet altijd even makkelijk om uit de pure machinecode te begrijpen wat er allemaal gebeurt. Voor de BIOS is dat probleem wel op te lossen: er bestaat een BIOS-listing met de originele assembler-programma's van Microsoft. Helaas is dat boek – MSX BIOS, The Complete MSX Basic I/O Listing, uitgegeven door Qest Publishing Inc in 1985 – duur en bijna onmogelijk te verkrijgen.

Wat blijft er dan over? Programma's van anderen. In MCM hebben we al ettelijke programma's gepubliceerd die hooks ombuigen, sommige netter dan andere. Deze keer een programma dat in wezen niets anders doet dan een hook omleiden: het is niet meer dan een skelet voor eigen programma's, die de CMD-opdracht willen gebruiken.

EIGEN COMMANDO'S MAKEN IN MSX-BASIC

Netjes buigen

De werking van de CMD is erg elegant – ontworpen met toekomstvisie, zou je bijna zeggen. Het idee is, dat meerdere programma's zichzelf in het geheugen van de computer kunnen installeren. Deze kunnen dan allemaal reageren op een bepaald CMD-commando. Samen vormen ze als het ware een keten.

Als MSX-Basic een CMD-opdracht tegen komt, wordt de CMD-hook op adres FE0D (hex) aangeroepen. Het programma dat de hook 'omgebogen' heeft, moet nu zelf onderzoeken of de juiste letters achter het CMD-commando staan. Als dat niet zo is, roept de routine gewoon weer de oude hook aan. Mocht die ook al omgebogen zijn, op het moment dat dit programma geïnstalleerd werd, blijft het oude programma ook actief. Een voorbeeld:

Direct na het opstarten van de computer wordt er een programma uitgevoerd, dat de CMD-hook ombuigt. Dit programma – A, bijvoorbeeld – reageert op: CMD A en doet dan iets. Op het moment dat programma A zichzelf installeert en de hook ombuigt, is de hook zelf nog inactief en bevat vijf RET-instructies. Programma A copieert deze vijf instructies naar een andere plaats in het geheugen. Als het commando:

CMD Z

nu moet worden uitgevoerd, roept MSX-Basic de CMD-hook aan. Die is omgebogen, dus programma A wordt aangeroepen. Die ziet, dat er geen A maar een Z achter het woord CMD staat en springt vervolgens naar de eerder bewaarde hook toe. Die bevat RET-instructies, dus programma A keert terug naar de Basic-ROM, die de boodschap 'Illegal function call' afdrukt. Stel nu dat er vervolgens een programma B opgestart wordt, dat op 'CMD B' reageert en dat zichzelf ook 'netjes' installeert. De CMD-hook wijst op programma A, dus als er:

CMD A

uitgevoerd moet worden, ziet B dat dit niet voor hem bestemd is, roept de hook aan en komt bij programma A terecht. Die constateert de letter A achter CMD en voert zijn taak uit.

Programma B heeft zichzelf dus als het ware tussen Basic en programma A ingevoegd. Dit kan net zo vaak gebeuren als het geheugen toelaat, mits elk programma zichzelf maar netjes installeert – en mits er natuurlijk geen twee programma's geladen worden die op hetzelfde CMD-commando reageren! Het ombuigen van de hook wordt verzorgd door de routine ZETHOOK in de listing van CMDEMO, waarop we zo nog terugkomen. De hook wijst, na ombuigen, op de routine START.

De CMD-hook

Hoe zit de CMD-hook nu precies in elkaar? Bij aanroep van de hook bevat A een karakter en HL een adres. Dit karakter is de eerste geldige letter na het woord CMD: spaties worden overgeslagen. In de praktijk wordt het door de Basic-ROM opgehaald met behulp van de BIOS-functie CHRGTR – van CHaRacter GeTteR? – op adres 010H. Omdat deze erg vaak door Basic

gebruikt wordt, is hij op een adres geplaatst dat met een RST-instructie kan worden aangeroepen – en RST-instructies zijn maar één byte lang in plaats van de drie bytes van een CALL-opdracht. De opdracht:

RST 010H

is dus gelijk aan:

CALL 010H

CHRGTR verwacht als invoer het HL-register, dat een adres in het Basic-programma moet bevatten. Vanaf dit adres worden spaties overgeslagen, er wordt een karakter gelezen en HL krijgt een nieuwe waarde. Daarnaast wordt de carry-vlag gezet als het karakter een cijfer is; de zero-vlag wordt gezet als het einde van een opdracht bereikt is. Dit kan betekenen dat de regel afgelopen is, maar ook dat er een dubbele punt achter de opdracht staat.

Wat moet een programma, dat de CMD-hook gebruikt, nu doen? Allereerst moet er onderzocht worden of het CMD-commando wel voor dit programma bestemd is. Daarvoor moet het programma zelf met RST 010H-opdrachten karakters lezen. Wordt er een ongewenst karakter gevonden, dan moeten HL en AF ongewijzigd blijven en moet de oude hook worden aangeroepen – misschien dat die er iets mee aan kan vangen.

Listing 1: CMDEMO

```
1Ø REM CMDEMO                                                        Ø
2Ø REM                                                               Ø
3Ø REM Demonstratie van het CMD-commando                             Ø
4Ø REM MSX/MS-DOS Computer Magazine                                  Ø
5Ø REM                                                               Ø
6Ø AD%=&HFA75                                                       91
7Ø READ A$: IF LEFT$(A$,1)<>"*" THEN POKE AD%,VAL("&h"+A$): AD%=AD%+1: GOTO 7Ø   14
8Ø DEF USR=&HFA75: A=USR(Ø)                                        1Ø6
9Ø PRINT "Druk op een toets voor een voorbeeld: ";: A$=INPUT$(1)   169
1ØØ FOR I%=1 TO 1ØØØ: CMD COLOR: NEXT I%: CMD BP                    34
11Ø END                                                            172
12Ø                    '                                           134
13Ø                    '         ORG    ØFA75H     ;programma in PLAY-rij 3    55
14Ø                    '                                           138
15Ø                    'CHGCLR   EQU    Ø62H       ;BIOS COLOR-routine     36
16Ø                    'BEEP     EQU    ØCØH       ;BIOS BEEP-routine     223
17Ø                    '                                           144
18Ø                    'FORCLR   EQU    ØF3E9H     ;voorgrondkleur        232
19Ø                    'BAKCLR   EQU    ØF3EAH     ;achtergrondkleur      1Ø4
2ØØ                    'HOOK     EQU    ØFEØDH     ;CMD-hook               61
21Ø                    '                                           133
22Ø                    'CMDLET1  EQU    "B"        ;eerste letter na CMD  189
23Ø                    'CMDLET2  EQU    "P"        ;tweede letter         125
24Ø                    '                                           139
25Ø                    'COLORTK  EQU    ØBDH       ;token voor COLOR      2Ø2
26Ø                    ';                                           23
27Ø                    '; Deze routine buigt een hook om naar een adres START   183
28Ø                    '; De oude hook wordt opgeborgen op adres OUDHK           53
29Ø                    ';                                           29
3ØØ DATA F3          : 'ZETHOOK  DI                ;even geen interrupts   18
31Ø DATA 11,CB,FA    : '         LD     DE,OUDHK   ;bewaar-plaats         122
32Ø DATA 21,ØD,FE    : '         LD     HL,HOOK    ;hook-adres            223
33Ø DATA Ø1,Ø5,ØØ    : '         LD     BC,5       ;een hook is 5 bytes    44
34Ø DATA ED,BØ       : '         LDIR              ;verplaats HOOK naar OUDHK   2Ø4
35Ø                    '                                           142
36Ø DATA 11,9Ø,FA    : '         LD     DE,START   ;adres START in DE     135
37Ø DATA 21,ØD,FE    : '         LD     HL,HOOK                       164
38Ø DATA 3E,C3       : '         LD     A,#C3      ;JP-instructie         18Ø
39Ø DATA 77          : '         LD     (HL),A     ;in 1e byte van de hook   35
4ØØ DATA 23          : '         INC    HL                              6Ø
41Ø DATA 73          : '         LD     (HL),E     ;leid de hook om naar   68
42Ø DATA 23          : '         INC    HL                              64
43Ø DATA 72          : '         LD     (HL),D     ;START                 19Ø
```

```
440 DATA FB          : '          EI                      ;interrupts weer aan          142
450 DATA C9          : '          RET                                                   254
460                    ';                                                                25
470                    '; Hier begint de CMD-afhandeling. A bevat de gelezen            181
480                    '; letter, HL de "tekstpointer"                                  137
490                    ';                                                                31
500 DATA F5          : 'START     PUSH  AF                ;bewaar alle registers         47
510 DATA C5          : '          PUSH  BC                                              192
520 DATA D5          : '          PUSH  DE                                              114
530 DATA E5          : '          PUSH  HL                                               72
540                    '                                                                142
550 DATA FE,BD       : '          CP    COLORTK           ;token van COLOR?             231
560 DATA 28,10       : '          JR    Z,INIT                                          171
570                    '                                                                148
580 DATA FE,42       : '          CP    CMDLET1           ;onze letter?                 121
590 DATA 20,05       : '          JR    NZ,RETURN                                        56
600 DATA D7          : '          RST   010H              ;haal volgende letter         114
610 DATA FE,50       : '          CP    CMDLET2                                         246
620 DATA 28,07       : '          JR    Z,INIT            ;twee letters gevonden        116
630                    '                                                                141
640 DATA E1          : 'RETURN    POP   HL                ;herstel tekstpointer          23
650 DATA D1          : '          POP   DE                ;en andere registers          236
660 DATA C1          : '          POP   BC                                              104
670 DATA F1          : '          POP   AF                ;en oude karakter              95
680 DATA C3,CB,FA    : '          JP    OUDHK             ;probeer volgende hook         62
690                    ';                                                                33
700                    '; Start CMD-commando:                                           223
710                    ';                                                                18
720 DATA D1          : 'INIT      POP   DE                ;oude tekstpointer             91
730 DATA E5          : '          PUSH  HL                ;vervang door nieuwe          143
740                    '                                                                144
750 DATA FE,BD       : '          CP    COLORTK           ;was het CMD COLOR?           199
760 DATA 28,05       : '          JR    Z,DOCLR           ;ja, spring verder            152
770                    '                                                                150
780 DATA CD,C0,00    : '          CALL  BEEP              ;nee. doe BEEP                209
790 DATA 18,11       : '          JR    EINDE                                           111
800                    '                                                                137
810 DATA 3A,EA,F3    : 'DOCLR     LD    A,(BAKCLR)        ;achtergrond naar A           220
820 DATA 67          : '          LD    H,A               ;en door naar H               128
830 DATA 3A,E9,F3    : '          LD    A,(FORCLR)        ;voorgrond naar A             173
840 DATA 32,EA,F3    : '          LD    (BAKCLR),A        ;en naar achtergrond          152
850 DATA 7C          : '          LD    A,H               ;oude achtergrond             198
860 DATA 32,E9,F3    : '          LD    (FORCLR),A        ;nieuwe voorgrond              64
870 DATA CD,62,00    : '          CALL  CHGCLR            ;zet de kleuren               161
880                    ';                                                                33
890                    '; Klaar: terug naar Basic                                       184
900                    ';                                                                18
910 DATA E1          : 'EINDE     POP   HL                ;(nieuwe) tekstpointer         93
920 DATA D1          : '          POP   DE                                               19
930 DATA C1          : '          POP   BC                                              101
940 DATA F1          : '          POP   AF                ;oude karakter                  6
950 DATA F1          : '          POP   AF                ;verwijder CMD-returnadres    253
960 DATA D7          : '          RST   010H              ;haal nieuw karakter          204
970 DATA C9          : '          RET                                                     7
980                    '                                                                154
990                    'OUDHK     DEFS  5                 ;5 bytes ruimte voor hook     223
1000 DATA *                                                                             215
```

```
1Ø GOTO 2Ø: COLOR                                                                              232
2Ø IF PEEK(&H8ØØC)=255 THEN PRINT "Token: "HEX$(PEEK(&H8ØØD))" (functie)" ELSE P
RINT "Token: "HEX$(PEEK(&H8ØØC))" (commando)"                                                  194
```

Listing 2: TOKEN

Aan de slag

Als wél aan alle eisen voldaan is, voert het programma zijn taak uit. Dan moet het programma alvast het volgende karakter lezen en terug springen naar Basic. Dat kan niet zomaar met een RET-instructie, omdat die immers terug zou springen naar de plaats in het ROM waar de hook werd aangeroepen — en het ROM reageert domweg met een 'Illegal function call'.
De juiste methode is, om het returnadres zelf van de stack te halen met een POP-opdracht en daarna pas de RET-instructie uit te voeren. Eén van de twee POP AF-instructies in de routine EINDE is hiervoor bedoeld.

BEEP en COLOR

Het voorbeeld-programma reageert op twee CMD-instructies:

CMD BP

en:

CMD COLOR

Het eerste doet niets anders dan BEEP; de tweede vorm keert de huidige voor- en achtergrondkleuren om. Beide allesbehalve wereldschokkend, maar het is dan ook maar een voorbeeld. Het controleren van de CMD-opdracht gebeurt in START. Hier worden eerst alle register-paren op de stack bewaard. Dan wordt er bekeken of het ingelezen karakter soms gelijk was aan COLORTK. Dit is een met EQU gedefinieerde waarde, die gelijk is aan 0BDH. (Basic 'tokeniseert' elke Basic-regel, waarbij sleutelwoorden door één byte vervangen worden. Het token voor COLOR is BD. We komen zo terug op een methode om het token van een sleutelwoord op te sporen). Als dit token gevonden wordt, springt START door naar INIT. Is het eerste karakter niet gelijk aan COLORTK, dan is het misschien een B, de eerste letter van de string "BP". Als dit ook niet zo is, gaat het programma verder bij RETURN, waar de registers worden hersteld en de oude hook wordt aangeroepen. Was de eerste letter wel een B, dan wordt er een volgende letter gelezen. Is dit een P, dan wordt INIT aangeroepen; in alle andere gevallen springt het programma door naar EINDE. INIT is heel eenvoudig: afhankelijk van het karakter in A wordt de BIOS-routine BEEP aangeroepen, of worden voor- en achtergrondkleur verwisseld met behulp van de BIOS-routine CHGCLR. INIT vervangt het bovenste woord op de stack, de oude tekstpointer, wel eerst door de nieuwe, die eventueel ontstaan is door het lezen van de tweede letter van "BP".
Bij EINDE wordt de — eventueel aangepaste — tekstpointer weer opgehaald, evenals de rest van de registers. Daarna wordt het return-adres van de stack gehaald, dat normaal gesproken terug zou keren naar de ROM-routine die de CMD-**hook** aanriep.
Maar die zal een foutmelding geven, zodat we liever terugspringen naar de ROM-routine die weer de CMD-**routine** aanriep. Deze beschouwt de CMD-opdracht als afgehandeld en gaat verder met de rest van het programma.

Tokens opsporen

Omdat MSX-Basic met tokens werkt, bevat elke MSX-computer een lijst met die tokens in het ROM. Het beginadres van die tokens is meestal hetzelfde, maar dat is niet gegarandeerd. Een 'quick and dirty' methode om het token bij een gereserveerd woord te vinden is het gebruik van TOKEN, zie listing 2.

De eerste regel van dat programma springt meteen door naar de tweede en lijkt dus zinloos. Maar het gaat om de rest van die regel: daar moet een sleutelwoord komen te staan, zoals PRINT, COLOR, KEY of TO. De tweede regel haalt met PEEK-opdracht het token op uit de eerste regel en drukt die hexadecimaal af. De eerste regel mag dus absoluut niet veranderd worden: hij moet precies zo worden overgenomen.

Door nu op de plaats van het woord COLOR een ander sleutelwoord in te vullen kan het token daarvan opgespoord worden. Let op: als het token gelijk is aan de ASCII-code van de eerste letter van het sleutelwoord, dan is er iets mis. Het sleutelwoord is misschien niet goed gespeld, of de eerste regel is niet — meer — in orde. Controleer eventueel zelf de tokens van PRINT (91), LIST (93), KEY (CC) en TAN (8D). De laatste is een functie; MSX-Basic behandelt die net als de andere tokens, behalve dat ze worden voorafgegaan door een FF-byte.

Uitbreiden

Dit voorbeeld-programma is op zich natuurlijk zinloos, maar het is een skelet dat zonder al te veel moeite uitgebreid kan worden tot een compleet programma dat verschillende CMD-commando's toevoegt aan Basic. Hoewel zulke CMD-opdrachten in een programma veel duidelijker zijn dan USR-aanroepen, zijn ze minder veelzijdig. De USR-opdracht is een functie, dat wil zeggen: de ML-routine kan een resultaat terugleveren. En een USR-functie kan een parameter aangeleverd krijgen. Dat laatste is met de CMD-opdracht ook mogelijk: in principe is er niets tegen om een CMD-opdracht te maken die tussen haakjes een serie parameters verwacht, net als een CALL-opdracht. Bijvoorbeeld:

CMD COLOR(10,1)

Maar het inlezen van 'evalueren' van die parameters is een vervelend karweitje, omdat commando's als:

CMD COLOR(&H25,1E-3)

of:

CMD COLOR(A%,SIN(23)*C)

eigenlijk ook goed afgehandeld moeten worden. Hiervoor zijn de nodige Basic-routines aanwezig in het Basic-ROM, maar die zijn — qua adres in het ROM — juist niet gegarandeerd voor alle MSX-versies...
Misschien dat we er desondanks nog een keer op terugkomen om CMD-opdrachten verder te kunnen verfijnen.
Maar zelfs in de 'simpele' vorm is het CMD-commando voor van alles en nog wat bruikbaar. We houden ons aanbevolen voor leuke en/of nuttige toepassingen!

I/O'tjes

I/O'tjes zijn kleine advertenties voor particulieren. Als u iets zoekt, of juist iets kwijt wilt, op computer-gebied, plaats dan een I/O'tje. Gebruik daarvoor de antwoordkaart uit dit blad. I/O'tjes zijn gratis voor abonnees, anderen betalen voor deze service slechts *f* 5,–. De redactie behoudt zich het recht voor I/O'tjes zonder opgaaf van redenen te weigeren. Gezien de omvang van het illegale copiëren zullen alle aanbiedingen van software – ook als dit samen met hardware gebeurt – worden geweigerd.
Slechts zelfgeschreven programma's mogen tegen een niet-commerciële prijs worden aangeboden.
Ook andere commerciële advertenties worden geweigerd, evenals I/O'tjes met een postbus- of antwoord-nummer.
Vermeld altijd uw volledige adres op de antwoordkaart, ook al wilt u slechts met uw telefoonnummer in deze rubriek worden opgenomen.

AANWIJZINGEN VOOR INZENDERS

Schrijf, in duidelijke blokletters, alleen binnen het aangegeven kader en vermeld daarin telefoon of adres. Alles wat buiten het kader valt wordt niet opgenomen. Vul het formulier vakje voor vakje in. Laat een vakje open (spatie) tussen de woorden laat alleen een vakje leeg als daar ook echt een spatie moet staan. Maak duidelijk onderscheid in hoofd- en kleine letters. Vergeet geen leestekens zoals punten en komma's.

INPUT

Leden voor ons blad MSX-journaal, per jaar *f* 30,-. Tel.: 01827-2272, Arjan, MSX GG Gouda.

PHILIPS MSX2. Type NMS 8280 in goede staat, p.n.o.t.k. Tel.: 05999-12544, Gerrit Dijkstra.

VS 0080 kl. mon. Tel.: 085-431665.

Een z.g.a.n. AMIGA 500 zonder bij behoren, max. *f* 200,-

Printer NMS 1431 tegen red. prijs, Liefst in regio Weert, Eindhoven, Venlo. Tel.: 04766-2103, Carlo.

PHILIPS NMS 8280. Tel.: 013-422382.

LEZERS ADVERTENTIES

Voor MSX nieuws belt u intersoft databank. Tel.: 043-470243, pagina 530, 24 uur online.

Contact met PC-gebruikers. Tel.: 08370-10376.

Facmanschap is meesterschap, FAC-demo 4. Sublieme Graphics en 10 kanaals muziek! MSX beurs Tilburg! Tel.: 04930-15638.

Wie kan mij aan het tekenprogramma Creatief Greetings van SONY helpen. Tel.: 01718-75457.

Programma of module voor een stamboom. Tel.: 08012-1850, R. Vos.

Wie heeft voor mij PTC print nr. 4 MSX gids 1 t/m 11, 14 en 15. MCM 1, 2 en 5. Ik bied *f* 6,- per nr. Tel.: 070-3210266.

MSX2 met dubb. zijdige drive en/of kl. mon., Tel.: 02963-1929, Floris.

MSX printer. Tel.: 020-456113.

MSX diskdrive 3.5 inch-720Kb. Tel.: 055-217736, na 18.30 uur , Marth.

FM-Pac MSX-Dos 2.20, 512Kb, Mem. Mapper eventueel ruilen met MSX printer (NMS1431). Tel.: 04782-571, Ward.

Wie kan de enkelzijdige drive in mijn VG8235 vervangen voor een dubbelzijdige? En een ROMBIOS boek. Tel.: 072-618764.

FM-Pac, MSX-Dos 2.20, 512Kb memory Mapper eventueel met MSX printer, NMS 1431. Tel.: 04782-571, Ward.

Matrix printer voor MSX 1 liefst in de omg. Gelderland. Tel.: 03417-60223, Egbert.

RS232C NMS 1210. Tel.: 05960-18053.

Autocad tekenprogramma voor MSX2 comp., SONY kl. plotter. Tel.: 05457-72703.

SONY 700 eventueel met toebehoren. Tel.: 074-663536

FM-Pac voor red. prijs. Tel.: 020-903382.

MSX printer liefst VW 0020. Tel.: 035-832334.

Listing BPut/Bget routines CUC journaal no. 19. Tel.: 01180-23568, na 17.00 uur.

Comp. PHILIPS NMS 8245. Tel.: 09-3211763649

Gez.: FM-Pac PANASONIC, PHILIPS muziek module NMS 1205. Tel.: 03497-3375, Sven.

Een Z80 assembler die op IBM compatible kan draaien, wie heet die, of wet die te koop. Tel.: 01827-5760.

SONY HBF-700P of D in een goede staat, met of zonder toebehoren voor max. *f* 400,-. Tel.: 05430-69380, na 14 mei 1990.

MSX 2, liefst PHilips, met diskdrive, datarec., tel.: 02984-3048 na 15.30 uur, Geert.

PHILIPS VS 0080 kl. mon. , i.z.g.s. met boeken/doos/snoeren etc. Tel.: 085-231694, na 18.00 uur.

Goed sorteer programma, voor de MSX, voor zelf geschreven platen programma's Tel.: 01745-12458.

Een FM-Pac. Tel.: 04705-2818.

Ik zoek een FM-Pac. Tel.: 080-561620, Mark.

MSX2 comp. 2 drives, klmon., evt. printer en overige toebeh. Evers, tel. 01820-39626, na 19h.

Een FM-Pac, p.n.o.t.k. Tel.: 058-129616.

PHILIPS MSX printer 1431 of 1421 in het noorden van het land. Tel.: 05909-1613.

Philips MSX2-computer, in goede staat! 8235 of hoger, met kl. monitor. Prijs max. *f* 850,-. Leon, tel. 08360-28041.

512K memory mapper t.e.a.b. Tel.: 043-645011, na 19.00 uur.

Defecte computers, geef redelijke prijs. Tel.: 03410-16183, na 18.00 uur.

FM-Pac t.e.a.b. Tel.: 02153-11844.

Toetsenbord Sony HBF-700P. Kruit, tel.: 010-4581600.

Ik zoek een MSX-printer. v. Ekeren, tel. 075-158028.

NMS8280, graphic tablet, robotarm. Ernst, 030-734942.

Diskdrive-interface voor MSX-computer. Alwin, tel. 053-331856.

FM-PAC voor max. *f* 50,-. Rogier, tel. 08360-28086 of Dirk, tel. 08362-27356.

Sony HBF700P. Omg. Friesland. Benedictus, tel. 01522-1845.

OUTPUT

Plotter, software op disk, handleiding. Tel.: 01827-2272, Arjan.

VSE databanken. Telefoon Apel.: 055-429450, A'dam: 020-972634, Dint.: 01672-2076, Eind.: 04929-64979, Veen.: 08385-29020, 1200/75 24 uur.

PHILIPS MSX-2 NMS 8235 inc. boeken, *f* 550,-. Tel.: 045-322839, na 17.00 uur.

PHILIPS NMS 8245, boeken, disks, 3 peeks & pokes boeken, MCM nr. 24 t/m 36, prijs *f* 850,-. Tel.: 040-815082, na 18.00 uur.

!!Aangeboden!!, MSX computermagazines nr. 5 t/m nr. 37, p.n.o.t.k. Tel.: 08306-43035.

T.k.: PHILIPS musicmodule, *f* 100,-. Tel.: 08380-16675, Jochem, Ede.

SONY HBF-700P MSX2, PHILIPS amber mon., nms 1421 NLQ printer, alles voor *f* 900,-. Tel.: 01720-22471.

MSX1 comp., datarec., joyst., *f* 285,-. Tel.: 071-221651.

T.k.: PHILIPS VW0020 MSX printer voor *f* 1800,-. Tel.: 05700-37164.

T.k.: MSX2 NMS 8250, printer, boeken, joyst., vraagprijs *f* 1750,-. Tel.: 08380-17413.

SONY HB-700P ANCONA 80 kleuren mon., MT-Telcom, modem EPSON GX80, MSX-print. Alles in prima staat, *f* 1600,-. Tel.: 08850-75946.

PHILIPS MSX2 NMS 8250, dik cursus boek. Tel.:070-3235821.

MSX1 ROMBIOS boek, *f* 25,-. Tel.: 072-618764.

MSX VG 8020 ACONA kl. mon., datarec., NMS 1421, printer, koopje. Tel.: 01193-410.

MSX PHILIPS monochrome mon., ƒ 149,-, B-diskdrive SONY HBD-30W, 720 Kb, ƒ 449,-. Tel.: 01751-78538.

MSX2, NMS 8250,kl. mon., printer VW 0030, joyst., muis, cass.rec., tekentablet. Vr.pr.: 08850-15022.

MSX2, NMS 8250, NMS 1421 printer, 32 disks, joyst., boeken, ƒ 1550,-. Ma.- Vr, na 19.00 uur, tel.: 080-566591, vraag naar Hugo.

PHILIPS MSX2 NMS 8280 nieuw in doos, 8 mnd. garantie, muis, joyst., lit., ƒ 1700,-. Tel.: 08897-71706.

NMS 1431 printer, linten. ƒ 575,-. MSX, muis, PHILIPS, ƒ 50,-. Tel.: 070-3239766.

TOSHIBA keyboard HX-MU900, MXMU 901, Tel.: 05190-2164.

Komt u ook naar de MSX-Beurs in Tilburg begin april?? Kom dan beslist eens een kijkje nemen bij de FAC!!

Draagbare mini mon. tv z/w, 12 cm. buis. Gewicht 2 kl.met schoudertas, ƒ 150,-. Tel.: 020-250345.

T.k. NMS 1421 PHILIPS MSX ƒ 400,-, z.g.a.n. NLQ, ong. 1 jr. oud. Tel.: 079-412163, 18.00 uur.

Printer NMS 1431, z.g.a.n. nog met eerste lint, ƒ 600,-. Tel.: 033-753661 na 18.00 uur.

MSX printer VW 0030 vr.pr. ƒ 575,-. Muis 3810 ƒ 80,-. Tel.: 02154-12123.

NMS 8235 tape rec., met boeken, z/w tv drive b. Prijs in één koop, ƒ 950,-. Tel.: 01890-15564.

CANON U-20, datarec., boeken, joyst., z.g.a.n., ƒ 395,-. Tel.: 08350-24737, na 18.00 uur.

ANCONA groen mon., ƒ 100,-. Datarec., ƒ 25,-. Tel.: 02159-11157.

PHILIPS MSX2 NMS 8255, muis, joyst., boeken, tijdschriften, 30 disks, ƒ 1250,- evt. z/w tv. Tel.: 074-664191.

PHILIPS VG 8020, datarec., switchbox, hoes, uitsluitend in één koop, p.n.o.t.k. Tel.: 01725-71225.

NMS 8220 comp., met SONY disk drive, 6 disks, datarec., prijs ƒ 1000,-. Tel.: 04704-3481, Roel.

MSX2 NMS 8255, mon., muis, 4 cartridges, 40 disks, joyst., veel boeken, ƒ 2000,-. Tel.: 01135-2259.

T.k.: PHILIPS 8250 mon. groen, printer, muziekmodule, boeken, diskbak met 60 disks, ƒ 1500,-. Tel.: 04104-79079.

MSX2, NMS 8255, 2 drives, kl. mon., printer, boeken, ƒ 2200,-. Tel.: 035-232686.

MSX2 VG 8235, mon., printer VW 30, boeken, disks, ƒ 1200,-. Tel.: 023-282747.

MSX1 CANON V20, PHILIPS mon. V80, datarec., nauwelijks gebruikt! In doos ƒ 350,-. Tel.: 08370-18714.

SV1-738 met ingebouwde diskdrive, mon, event. met NMS 1431 printer, tel.: 04782-571, na 18.00 uur Waldo.

PHILIPS NMS 8245, datarec., joyst., kl. mon., VS 0080, handboek, ƒ 1600,-. Tel.: 02995-1843.

SVI-738 met ing. diskdrive, mon., eventueel met MSX printer. Tel.: 04782-571, na 18.00 uur, Ward.

MSX2 comp., NMS 8250 ƒ 850,-. SONY C41 kleuren printer/plotter incl. adapter, kabel, ƒ 300,-. Tel.: 020-954059.

MSX 2 HBF-9P, datarec., 12 cass., alles in één koop ƒ 450,-. Tel.: 01810-15720, na 18.00 uur.

SONY HB-201P MSX1, datarec., boeken, ƒ 700,-. Tel.: 020-363456.

NMS 8245 MSX2 ingebouwde diskdrive 3,50 inch, boek./tijdschriften, in verp. ƒ 850,-. Tel.: 020-843157, na 18.00 uur.

SONY HB-201, SONY CN-14 printer/plotter, Sanyo cass.rec., één koop ƒ 700,-. Tel.: 020-422875, na 17.30 uur.

NMS 8250, VW 0030, muziek module 1205, muis SBC-3810, PBNA cursus. Incl. boeken en 50 disks, ƒ 2000,-. Tel.: 08370-12446.

Dubb. PHILIPS RS232C interface RS 1211, vr. pr., ƒ 200,-. Tel.: 05905-5560, na 18.00 uur.

PHILIPS NMS 8250, printer 1421, muis, stofhoes, handleiding, voor ƒ 1200,-. Tel.: 01172-2736.

PHILIPS NMS 8220, SONY drive HBD-50, rec., graph, tabl., muis, NMS 1255, boeken, tijdschriften, vr.pr., ƒ 1150,-. Tel.: 03450-16280.

PHILIPS VG 8020, datarec., joyst., tijdschr., boeken, vr.pr. ƒ 295,-. Tel.: 03450-16280.

NMS 8245, MSX2 256Kb, in prima staat, 1,5 jaar oud, dubbelzijdig disk. (mon. ƒ 550,-.). Tel.: 070-3979679, Martijn.

MSX1 met aantal boeken en datarec., joyst., ƒ 215,-. In één koop ƒ 400,-. Tel.: 05998-34266, na 17.00 uur.

MSX2 computer, NMS 8250 nieuw incl. boeken. Tel.: 01879-1745, BGG 01878-3019.

PHILIPS VG 8020, defect toetsenbord, vr.pr., ƒ 90,-. (evt. in A'dam af te halen). Tel.: 05776-1841..

PHILIPS MSX 2 NMS 8220, datarec., 70 spelletjes, utility's, 33 tijdschriften, boeken, ƒ 600,-. Tel.: 02209-2837.

PHILIPS MSX2 comp. NMS 8250 met kl. mon., VS 0080, met printer VW 0030, datarec., NMS 1510 met spelletjes, boeken, enz.., ƒ 1950,-. Alles in één koop. Tel.: 020-424214

SONY trackball, type 68-7, ƒ 80,-. SANYO datarec., type DR 202A, ƒ 90,-. Tel.: 020-903382.

NMS 8250 kl. mon., VS 0030 HD2 MSX-Dos, boeken, ƒ 1800,-. Tel.: 03480-18130.

MCM's nr. 35 t/m 37 in een koop ƒ 100,-. T. Ritzer, T. Sybrandthof 35, 2433 GK, Utrecht.

PHILIPS MSX NMS 8245, kl. mon., muis, joyst., boeken, 90 disks, ƒ 1250,-. Tel.: 05660-1972.

NMS 8245 MSX2, boeken, MCM tijdschr., disks, kabel, 3 PPT boeken, stofhoes, prijs ƒ 700,-. Tel.: 040-815082, Bram.

VG 8020, rec., handl., ƒ 250,-. Tel.: 04709-2589, Horst(L).

PHILIPS G-7000 videopac, kleine reparatie noodzakelijk, 6 spellen, ƒ 175,-. Tel.: 03497-3375.

VY0011 B-drive in perfecte staat, ƒ 300,-, bellen na 17.00 uur of in het weekend. Tel.: 01621-16815, Aske Hopman.

NMS 8250 256K MSX2, kl. mon., datarec., music module NMS 1205. Tel.: 05114-2929.

SVI 728 met SVI 707 d.d. en SVI 727 80 column cartridge CP/M, PHILIPS mon., ƒ 450,-. Tel.: 03402-37448.

Printer CANON T-22A z.g.a.n. met veel papier, ƒ 200,-. Tel.: 03210-13493.

SONY RS232C interface voor MSX ƒ 175,-. Tel.: 08855-77323.

NMS 8245 incl. boeken, ƒ 800,-. NMS 1431 printer, ƒ 450,-. Tel.: 020-954059.

NMS 8280, NMS 1431 printer VS 0080, monitor, NMS 1150, tekenbord, compl. ƒ 3000,-. Tel.: 01110-16039.

PHILIPS MSX2 NMS 8250, PHILIPS printer NMS 1421, veel boeken, 2 joyst., ƒ 1750,-. Tel.: 015-123720.

MSX2 NMS 8250, 60 disks, 30 MSX comp. mag., veel handl. en boeken, 2 joyst., datarec. Tel.: 01612-27087.

MSX2 NMs 8250, printer VW 0030, 40 disks, datarec., tapes, kl. mon. VS 0080, i.z.g.s. t.e.a.b. Tel.: 023-287197.

SPELLING CHECKER

De ACME Software Company presenteert: De Spelling Checker voor MSX! Nooit meer woorden verkeerd gespeld. Handig voor het schrijven van brieven, onmisbaar voor sollicitaties, opstellen of scripties.

De Spelling Checker is universeel toepasbaar, het programma werkt ondermeer met TASWORD-2, WORDSTAR, EDIT en verder alle tekstverwerkers welke normale ASCII-files gebruiken.

f 95,- bij de MSX Specialist.

03410-26017

INVOER CONTROLE PROGRAMMMA

Zelfs de meest zorgvuldig geproduceerde en gedrukte listings sluiten niet uit dat er toch een fout kan worden gemaakt bij het intikken. Verwisselde cijfers of verkeerde leestekens leiden in het beste geval tot een foutmelding. Erger nog is het als een programma slechts schijnbaar goed, althans zonder fouten die de computer zelf kan bespeuren, werkt.

Om u te helpen dit soort problemen te voorkomen publiceert MSX Computer Magazine bij alle listings een controlegetal oftewel checksum per programmaregel. Achter iedere programmaregel staat een checksum. Om deze te vergelijken met uw zelf ingetikte programma dient het bijgaande Invoer Controle Programma, kortweg ICP. Met MCM is het Invoer Controle Programma het laatste programma dat u zonder hulp moet intikken!

Gebruiksaanwijzing ICP/6

ICP/6 berekent voor iedere ingetikte programmaregel een checksum, zodra u op de enter of return drukt. Deze checksum verschijnt dan linksonder op uw beeldscherm, op de positie waar anders de bij de F1 behorende tekst – COLOR – staat. Deze waarde moet overeenkomen met het getal dat in de listing bij de betreffende regel is afgedrukt, als dit niet het geval is heeft u een foutje gemaakt bij het intikken. In dat geval moet u de betreffende regel meteen verbeteren met behulp van de normale edit-mogelijkheden van uw MSX computer.
U hoeft de regel dus niet opnieuw in te tikken, ICP/6 kijkt altijd naar de hele programmaregel zoals die op het scherm staat, niet alleen naar wat er echt ingetikt wordt. Daardoor kunt u ook al eerder ingetikte programma-regels makkelijk controleren. Gewoon de regel listen, dan de cursor weer omhoog brengen tot deze zich ergens in de te checken programmaregel bevindt en op return of enter drukken. ICP/6 maakt natuurlijk onderscheid tussen hoofd- en kleine letters en dat kan soms problemen opleveren. Bij het intikken van een programma zult u meestal de Basic woorden in kleine letters intikken, maar bij het listen van een regel verschijnen ze juist wel in hoofdletters. ICP/6 gaat er van uit dat Basic woorden met hoofdletters geschreven moeten worden, net zoals ze in de listings staan. Zet daarom de Caps-lock op uw machine – waardoor alle letters hoofdletters worden – aan. Alleen als er gewone letters in de listing staan moet u die Caps-lock even uitzetten. Wat natuurlijk ook kan is de regel intikken, op return drukken, dan de zojuist ingevoerde regel opnieuw listen met LIST., de cursor weer in die regel plaatsen en nogmaals op return drukken. De tweede keer 'kijkt' ICP/6 naar de geliste regel en daar heeft uw MSX keurig alle Basic woorden in hoofdletters vertaald. Voor REM-regels (die ook met het ' teken aangegeven kunnen worden) wordt de checksum op nul gesteld. Slechts als de REM of het '-tekentje niet meteen na het regelnummer staan tellen deze regels wel mee.
ICP/6 is in feite een machinetaal-programma. De Basic-listing zet deze ML – die in de DATA staat – op de juiste plek in het geheugen, waarna het zichzelf weer uitwist. Als u na het runnen van ICP/6 een LIST-commando geeft, dan zult u zien dat het programma schijnbaar helemaal verdwenen is. De machinetaal echter staat ergens hoog in het geheugen voor u klaar, zoals u kunt zien aan de getallen links-onder. Die ML kunt u eventueel uitzetten door de F1 in te drukken, en weer aanzetten middels het commando:

A = USR(0)

Om de ML helemaal te verwijderen moet u de MSX resetten of even uitschakelen. Lange listings intikken kost vaak meer dan één zitting, als u stopt moet u datgene wat u tot dan toe ingetikt heeft gewoon eventjes saven, zoals u altijd een Basic-programma wegschrijft. Alleen de Basic-tekst wordt dan bewaard, niet het ICP/6. Om de volgende keer weer door te gaan dient u eerst ICP/6 weer te laden en te runnen, waarna u het programma waar u aan bezig was weer kunt laden. Een kind kan de was doen!

ICP VOOR FOUTLOOS INTIKKEN VAN LISTINGS

Hoe de listings in te tikken

MSX Computer Magazine publiceert alleen programma's die door de redactie uitgebreid getest zijn op hun deugdelijkheid. Om te voorkomen dat er bij het zetten alsnog fouten insluipen drukken we dat geteste programma vervolgens rechtstreeks af, met de laserprinter. De zo gemaakte listings zijn van een speciaal formaat, dat ontworpen is om fouten tijdens het intikken zoveel mogelijk te voorkomen. In programma's is iedere letter, ieder cijfer en elk leesteken van belang. De kleinste vergissing bij het intikken kan desastreuze gevolgen hebben. Om verwarring tussen de hoofdletter 'O' en het cijfer '0' te vermijden is de nul altijd doorgestreept. Let ook op het verschil tussen het cijfer '1' en de kleine letter 'l'.

Per regel drukken we precies zoveel tekens af als u op uw scherm ziet onder het intikken. Op MSX2 zijn dat er 80, op MSX1 37. Programmaregels die langer zijn worden afgebroken, net zoals dit op het beeldscherm van uw computer gebeurt. De getallen die in een aparte kolom rechts naast de eigenlijke listing staan moet u niet intikken, dit zijn de controle-getallen die samen met het Invoer Controle Programma u het mogelijk maken om een listing in één keer foutloos in te tikken.

Een veel voorkomende fout tijdens het intikken is het vergeten van de RETURN, die na iedere programmaregel moet worden ingetikt. Ook als de vorige regel precies 80 – of 37, bij MSX1 – tekens lang is, zodat de cursor al vooraan de volgende regel staat, is dit absoluut noodzakelijk. Een voorbeeld hiervan kunt u in het ICP, zelf zien; regel 180 is precies 37 tekens lang. Als u nu de RETURN vergeet, dan zal het lijken of regel 190 gewoon in het programma staat. Bij het listen verschijnt deze regel normaal.
U kunt 190 dan echter niet apart listen, want volgens de computer maakt die regel gewoon deel uit van regel 180, een commentaar-regel. Dit leidt uiteindelijk tot een foutmelding:
NEXT WITHOUT FOR IN 260
Want pas in regel 260 ontdekt de computer dat er een FOR-commando ontbreekt, omdat regel 190 niet als zelfstandige regel in de computer is ingevoerd.

ICP/6

ICP/6 is bijna helemaal gelijk aan de vorige versie, we hebben slechts een paar schoonheidsfoutjes verbeterd. Wie al een goed werkende exemplaar van een van de vorige versies bezit hoeft ICP/6 natuurlijk niet alsnog in te tikken.
Voor wie het weten wil, er zijn twee verschillen tussen deze versie en de vorige.

Zo bleek uit lezers-vragen dat men soms totaal geen checksums te zien kreeg, een op het eerste gezicht onverklaarbare fout. Pas na enig doorvragen bleek er een logica in het voorkomen van dat probleem te vinden, namelijk dat het alleen op MSX2 gebeurde. Toen was de oplossing snel gevonden. Immers, op een MSX2 kan men de scherm-instellingen vastleggen, zodat die min of meer permanent worden? En één van die instellingen is het afzetten van de functietoets teksten, inclusief het 'COLOR', waar ICP zijn checksums zet.
Kortom, ICP/6 zet altijd het functietoetsen display aan, zodat de checksums ook werkelijk te zien zijn.
De tweede wijziging in ICP/6 is nog eenvoudiger, we kwamen er namelijk achter dat de ingebouwde extra controle op de machinetaaldata niet helemaal goed was. In regel 210 vermenigvuldigen we nu de ASCII-waarde met de waarde teller + 1. Want aangezien die teller als startwaarde 0 heeft werden in ICP/5 de eerste bytes niet gecontroleerd!

Maar goed, u hoeft ICP/6 niet in te tikken, als u al een werkende versie van één van de eerdere ICP's heeft. Als het eenmaal werkt is er tussen ICP/3, ICP/4, ICP/5 en ICP/6 geen verschil.
ICP/2 echter is een stuk trager, hoewel het dezelfde checksums gebruikt.

```
1Ø ' ICP/6 voor MSX                              Ø
2Ø '                                             Ø
3Ø ' MSX/MS-DOS Computer Magazine                Ø
4Ø ' copyright MBI Publications B.V.
1985                                             Ø
5Ø '********************************             Ø
6Ø ' PAS TOP OF BASIC MEMORY AAN ****            Ø
7Ø CLEAR 2ØØ,(PEEK(&HFC4A)+256*PEEK(&
HFC4B))-2Ø7                                      43
8Ø B=PEEK(&HFC4A)+256*PEEK(&HFC4B): D
EFUSRØ=B+77                                      18
9Ø ' STEL SCHERM IN ****************             Ø
1ØØ SCREEN Ø: WIDTH 37: COLOR 15,4,4:
 KEY ON                                          78
11Ø ' GEEF INSTRUCTIE OP SCHERM *****            Ø
12Ø LOCATE 5,Ø: PRINT "MSX/MS-DOS COM
PUTER MAGAZINE"                                  149
13Ø LOCATE 5,2: PRINT "INVOER CONTROL
E PROGRAMMA/6"                                   67
14Ø LOCATE Ø,5: PRINT "Dit programma
maakt het mogelijk om  de listings ui
t dit blad foutloos in te voeren."               244
15Ø PRINT "Bij het intikken van progr
amma-regelsverschijnt, nadat u op 'RE
TURN' of   'ENTER' gedrukt heeft een
getal linksop de onderste regel."                21Ø
16Ø PRINT "Dit getal moet gelijk zijn
 aan de bijde listing afgedrukte chec
ksumwaarde.Als dit niet zo is, dan is
 er een    fout gemaakt bij het intik
ken."                                            35
17Ø PRINT "Let op, Basic-woorden moet
en met      hoofdletters geschreven wo
rden!"                                           223
18Ø ' INSTALLEER MACHINECODE ********            Ø
19Ø FOR R=Ø TO 2Ø6                               143
2ØØ READ A$                                      247
21Ø CS=CS+(ASC(LEFT$(A$,1))+ASC(RIGHT
$(A$,1))*2)*(R+1)                                97
22Ø IF LEFT$(A$,1)<>"*" THEN POKE B+R
,VAL("&H"+A$): GOTO 26Ø                          126
23Ø IF A$="**" THEN READ A$: AB=B+VAL
("&H"+A$): POKE B+R,AB-(INT(AB/256)*2
56): R=R+1: POKE B+R,INT(AB/256): GOT
O 26Ø                                            78
24Ø IF A$="*1" THEN READ A$: AB=B+VAL
("&H"+A$): POKE B+R,AB-(INT(AB/256)*2
56): GOTO 26Ø                                    24
25Ø IF A$="*2" THEN POKE B+R,INT(AB/2
56): GOTO 26Ø                                    168
26Ø NEXT R                                       46
27Ø ' CONTROLEER DATA-WAARDES *******            Ø
28Ø IF CS<>3396989# THEN CLS: PRINT "
U heeft een fout gemaakt in de data-
regels!": PRINT: PRINT "Eerst verbete
ren!": STOP                                      255
29Ø ' ZET CHECKSUMROUTINE AAN *******            Ø
3ØØ A=USRØ(Ø)                                    16
31Ø PRINT: PRINT "Begint u maar met i
ntikken"                                         212
32Ø NEW                                          44
33Ø ' MACHINECODE ******************             Ø
34Ø DATA 21,5E,F5,7E,23,FE,2Ø,2Ø,FA,7
E,23,FE,2Ø,28,FA,FE,27,28,D,FE,52,2Ø,
D,7E,FE,45,CØ,23,7E,FE,4D,CØ,E1,C3,**
,9F,FE,72,CØ,7E                                  61
35Ø DATA FE,65,CØ,23,7E,FE,6D,28,EF,C
9,36,27,1,6,Ø,21,**,47,11,7F,F8,ED,BØ
,3E,C9,32,DB,FD,C3,**,CB,63,6F,6C,6F,
72,2Ø,1,6,Ø                                      215
36Ø DATA 21,7F,F8,11,**,47,ED,BØ,21,7
F,F8,36,27,21,83,F8,36,27,23,36,D,21,
DB,FD,36,C3,23,36,*1,71,23,36,*2,C9,1
E,Ø,CD,**,ØØ,21,5E                               6
37Ø DATA F5,6,1,7E,4F,FE,27,28,B1,18,
1,4E,3E,Ø,B9,28,16,C5,16,7,CB,39,3Ø,1
,8Ø,15,28,4,CB,2Ø,18,F4,83,5F,C1,4,23
,18,E4,6B                                        47
38Ø DATA 11,8Ø,F8,26,Ø,1,64,Ø,CD,**,B
9,1,A,Ø,CD,**,B9,1,1,Ø,CD,**,B9,18,12
,37,3F,3E,Ø,ED,42,FA,**,C5,3C,18,F8,C
6,3Ø,12                                          57
39Ø DATA 13,9,C9,CD,C9,Ø,C9                      12
```

OEPS

We doen ons uiterste best om een foutloos blad te produceren. Toch is een tijdschrift ook maar mensenwerk en dat geldt zeker voor de programma's die erin staan. Dus sluipen er soms fouten in. Vandaar deze vaste rubriek, oeps, waarin we niet alleen fouten rechtzetten, maar ook verbeteringen zullen publiceren van eerder verschenen programma's en andere zaken.

EHBO MCM 36

In MSX Computer Magazine 36 is er foutje geslopen in de EHBO-rubriek, waar men het over de 'Back to the basics'-Poke heeft. In de tweede kolom op pagina zes, tweede alinea, beweert de schrijver dat het beginadres van de stack bij systemen met één drive op &H1AF begint. Pure nonsens natuurlijk, die stack staat een heel stuk verderop in het geheugen. Om precies te zijn, op &H1EAF. En dat scheelt een slok op een borrel, beste EHBO'er!

DiskView patchen

DiskView, uit MSX Computer Magazine nummer 31, is een prima diskmonitor. Maar toch wist onder andere lezer Marc van Iersel het programma op een foutje te betrappen. Als men namelijk middels de F8 – functie NewDisk – een disk van een ander type wilde gaan bekijken ging het fout. Stel dat men eerst met een enkelzijdige disk aan het werk was, dan werd een dubbelzijdige disk verkeerd geïnterpreteerd. Zulks tot verbazing van de programmeur, Steven van Loef, die toch echt eerst sector 0 las gevolgd door het – via de DOS-functie Get Allocation – inlezen van de diskparameters. En toch ging dat fout, hetgeen doet denken aan een slordigheidje in de disk-ROM. Blijkbaar 'denkt' de diskcontroller op dat moment, wanneer de disk nog draait na het lezen van sector 0, dat de originele disk nog steeds in de drive steekt. Ietwat eigenmachtig wordt het verzoek nieuwe diskparameters te lezen gewoon genegeerd. Wie zei daar dat computers geen eigen leven kunnen leiden?
Hoe dan ook, door die beide zaken om te draaien, dus eerst de parameters opvragen en dan pas sector 0 lezen, gaat de diskwissel wel goed. En dat kan men bereiken met het listinkje bij deze Oeps, dat DiskView patcht.
Even opletten: u moet dit programma goed controleren – bij voorkeur met het Invoer Controle Programma – alvorens het te runnen, met een kopie van DiskView in de A-drive. Die kopie wordt vervolgens aangepast, waarna even uitproberen van de verbeterde versie een goed idee is alvorens uw oorspronkelijke DiskView buiten gebruik te stellen.

Achter de schermen...

In MCM 37 stond een aardig artikel, met als naam 'Achter de schermen van MSX'. Jammer genoeg geeft dit verhaal echter ook een onbedoelde inkijk achter de schermen van de redactie.
Dat er een paar fouten in voorkomen, het zij zo, maar dat de listings volstrekt onmogelijk zijn, dat valt toch op. Betrapt, want de redacteur in kwestie was wat lui en heeft de laatste wijzigingen op zijn PC aangebracht, in een editor-programma. Niet getest, met andere woorden. We hebben de schuldige bestraffend toegesproken, in de laatste redactievergadering.
Zo blijkt in listing 2 regelnummer 20 twee keer gebruikt te zijn, waardoor ook de checksum niet meer klopte. Even aanpassen, tot de regelnummers van 10 tot en met 70 lopen, en het leed is geleden. In listing 3 zijn in regel 40 de haakjes rond het argument achter de functie VPEEK vergeten. Oftewel:

40 V = VPEEK(ST + 65*8 + I)

Listing 4 is ook al mis, in regel 30 moet natuurlijk staan:

30 FOR I = 0 TO 7

In de tekst zelf tenslotte gaat de schrijver de fout in op bladzij zes, rechtsonder, waar hij beweert dat het schermtype 0 32 bij 40 tekens telt. BASE(0) geeft inderdaad een tabel-adres van een tabel met 960 eenheden, maar die 960 worden wel bereikt doordat het scherm 24 bij 40 tekens telt. Gelukkig volstond lezer R.H. Kruger uit Graauw na deze waslijst met de opmerking dat het 'voor de rest een interessant artikel' was.
In diezelfde brief stelde deze lezer echter ook dat allerlei listings die we publiceren ten onrechte het GOTO-commando bevatten. IF voorwaarde THEN GOTO regelnummer, dus. Inderdaad, die GOTO is daar niet vereist. Maar we kiezen er specifiek voor om GOTO – voor de leesbaarheid – op zulke plekken wel te gebruiken.

```
1Ø REM DiskView patch                                                   Ø
2Ø OPEN "diskview.com" AS 1 LEN=1: FIELD 1,1 AS P$                      185
3Ø FOR N=1 TO 33: READ A$: T=T+(VAL("&H"+A$)*N): NEXT N: IF T<>22174 THEN PRINT
"FOUT IN DATA"                                                          193
4Ø RESTORE                                                              223
5Ø LSET P$=CHR$(Ø): PUT 1,1Ø15: PUT 1: PUT 1: PUT 1,2Ø94: PUT 1: PUT 1  224
6Ø I=1557                                                               123
7Ø READ D$: IF D$="**" THEN GOTO 8Ø ELSE LSET P$=CHR$(VAL("&H"+D$)): PUT 1,I: I=
I+1: GOTO 7Ø                                                            135
8Ø CLOSE 1                                                              2Ø9
9Ø PRINT "DiskView werkt nu goed met alle diskformaten": END            215
1ØØ DATA 3A,45,18,3C,5F,ØE,1B,CD,CØ,Ø1,32,41,18,22,42,18,CD,6F,ØA,ØØ,ØØ,ØØ,ØØ,ØØ
,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,2A,DC,Ø1,**                                          2Ø6
```

MSX CENTRUM

W. de Withstraat 27 1057 XG AmsterdamFax 167058
Tel. 020-167058 (bel voor informatie tussen 14.00 en 18.00 uur)

SUPER LAGE AANBIEDING MSX2 +



995,-

Wegens het vervallen van een export order naar Canada van Sanyo 2 + computers;
300 stuks type D en 75 stuks type 70 DD.
Onder andere de volgende gegevens:
Uitvoering 220 V, RBG uitgang, SCART voor monitor en T.V., Dubbelzijdige drive,
80K basic 3, 16K diskbasic, 16K TURBO BASIC 128 JIS (onder andere alle wereld-
talen), pauzetoets.

Aflevering vindt plaats in volgorde van bestelling. De levertijd bedraagt drie weken. De toestellen worden geleverd met Nederlandse importeur garantie en gebruiksaanwijzing.
Gemakkelijke betaling.

Bestel door overmaking van ƒ 995,- of door overmaking van ƒ 495,- en betaal restant bij aflevering onder rembours (510,-)

Per giro: 2922 NMB Amsterdam t.g.v. 69.59.15.592
Per bank: NMB Amsterdam rekening nr. 69.59.15.592

Naam: ....

Adres: ....

Postcode en Woonplaats: ....



HOMESOFT

Home Software Benelux bv
Küppersweg 63-65
2031 EB Haarlem
Tel.: 023 - 311 241
Fax: 023 - 318 488

Alle Homesoft producten worden geleverd via de erkende computer vakhandel
Dealer aanvragen welkom